



TESIS - IF185401

**PENGEMBANGAN MEKANISME *CHANGE  
DETECTION* UNTUK EFISIENSI ENERGI PADA  
*WIFI-BASED INDOOR POSITIONING SYSTEM***

**ADE HENDI**  
NRP. 05111850010032

Dosen Pembimbing  
Waskitho Wibisono, S. Kom., M. Eng., Ph. D.  
Ary Mazharuddin Shiddiqi, S. Kom., M. Comp. Sc., Ph. D

Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
2020



# LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
**Magister Komputer (M.Kom.)**

di

**Institut Teknologi Sepuluh Nopember Surabaya**

oleh:

**ADE HENDI**

**NRP: 05111850010032**

Tanggal Ujian : 30 Desember 2019

Periode Wisuda : Maret 2020

Disetujui oleh:

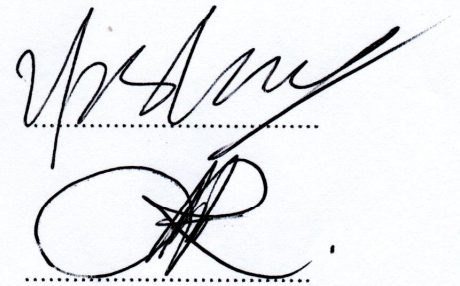
**Pembimbing:**

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

NIP. 197410222000031001

Ary Mazharuddin Shiddiqi, S. Kom., M. Comp. Sc. Ph. D.

NIP. 198106202005011003



**Penguji:**

Dr. Eng. Radityo Anggoro, S. Kom., M. Sc.

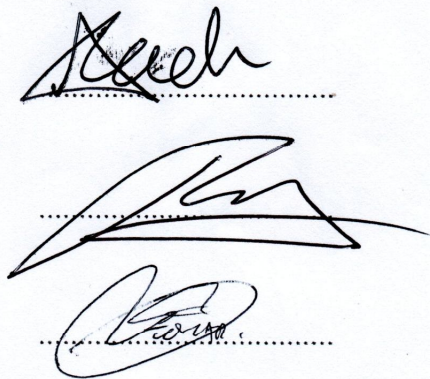
NIP. 1984101620081210002

Royyana Muslim Ijtihadie, S. Kom., M. Kom., Ph. D.

NIP. 197708242006041001

Tohari Ahmad, S. Kom., MIT., Ph. D.

NIP. 197505252003121002

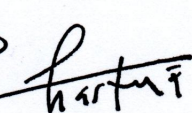


Kepala Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas



**Dr. Eng. Chastine Fatichah, S. Kom., M. Kom.**

NIP. 197512202001122002





**PENGEMBANGAN MEKANISME CHANGE DETECTION UNTUK  
EFISIENSI ENERGI PADA WIFI-BASED INDOOR POSITIONING  
SYSTEM**

Nama : Ade Hendi  
NRP : 05111850010032  
Pembimbing : Waskitho Wibisono, S.Kom., M.Eng., Ph.D.  
: Ary Mazaharudin Shiddiqi, S.Kom., M.Comp.Sc.,  
Ph.D.

**ABSTRAK**

Pengembangan mekanisme *change detection* mempunyai peranan penting terhadap Indoor Positioning System (IPS). Namun permasalahan yang masih umum dijumpai adalah konsumsi energi yang tinggi, karena proses WiFi *scanning* berjalan secara terus menerus. Proses WiFi *scanning* mengirimkan data dari klien ke server secara terus menerus, terkadang memberikan informasi yang sama dan berulang kepada user. Informasi yang dikirim secara redudan bisa berdampak pada konsumsi energi yang tinggi. Penelitian ini mengusulkan mekanisme perbaikan dengan *change detection* untuk penghematan energi dalam melakukan *sampling* secara adaptif pada kekuatan sinyal WiFi dengan *accelerometer* sebagai trigger. Mekanisme *change detection* yang dilakukan adalah mengukur nilai pada *accelerometer* dengan menentukan silent zone. Silent Zone merupakan rentang nilai yang didapatkan ketika *accelerometer* dalam kondisi diam. Apabila diketahui nilai kekuatan sinyal pada *accelerometer* melebihi nilai silent zone, maka diidentifikasi user dalam kondisi bergerak dan secara otomatis proses WiFi *scanning* akan berjalan. *Change detection* dengan *Bluetooth* mempunyai proses yang sama dengan menggunakan *accelerometer*. Algoritma yang diusulkan dapat menghasilkan penghematan daya baterai sebesar 4,384% untuk *scanning* dengan *change detection* menggunakan *accelerometer* dan 2,666% untuk *change detection* menggunakan *Bluetooth*.

**Kata kunci:** indoor positioning system, change detection, accelerometer, bluetooth, sampling secara adaptif.

# CHANGE DETECTION MECHANISM FOR ENERGY EFFICIENCY IN WIFI-BASED INDOOR POSITIONING SYSTEM

Name : Ade Hendi  
StudentIdentityNumber: 05111850010032  
Supervisor : Waskitho Wibisono, S.Kom., M.Eng., Ph.D.  
: Ary Mazaharudin Shiddiqi, S.Kom.,  
M.Comp.Sc.,Ph.D.

## ABSTRACT

The development of change detection mechanism has an important role in the Indoor Positioning System (IPS). In IPS technology, a lot of battery power will be used because the WiFi scanning process runs continuously. The WiFi scanning process sends data from the client to the server continuously, sometimes providing the same and repeatable information to the user. Information sent redundantly can have an impact on high energy consumption. In this research, the researchers developed a repair mechanism with change detection to save energy in an adaptive sampling of the strength of the WiFi signal with the accelerometer as a trigger for the adaptive process. Change detection mechanism that is done is measuring the signal strength on the accelerometer by determining the silent zone. Silent Zone is the range of values obtained when the accelerometer is at rest. If it is known that the signal strength value on the Accelerometer exceeds the value of the silent zone, the user is identified in a mobile condition, and the WiFi scanning process will automatically run. Change detection with Bluetooth has the same process using an accelerometer. The algorithm we propose can produce a battery-saving of 4.384% for scanning with change detection using an accelerometer and 2.666% for change detection using Bluetooth.

**Keywords:** indoor positioning system, change detection, accelerometer, bluetooth, adaptive sampling.

*[Halaman ini sengaja dikosongkan]*



## KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Allah SWT, atas segala rizki, berkah, nikmat serta karunia-Nya yang terlimpahkan kepada penulis, sehingga penulis akhirnya dapat menyelesaikan penelitian dengan judul “Pengembangan Mekanisme Change Detection Untuk Efisiensi Energi Pada Wifi-Based Indoor Positioning System”.

Penulis juga ingin mengucapkan banyak terimakasih kepada teman-teman seangkatan yang tidak perlu disebutkan namanya satu-satu. Disadari sepenuhnya bahwa tanpa bantuan dari berbagai pihak, penelitian ini tidak akan terselesaikan dengan hasil seperti sekarang ini. Oleh karena itu pada kesempatan ini penulis ingin menyampaikan terima kasih yang sebesar-besarnya, kepada:

1. Allah SWT atas limpahan nikmat iman, islam, kesehatan, waktu, serta berbagai kemudahan dari arah yang tidak pernah diduga sebelumnya, sehingga penulis dapat menyelesaikan penelitian ini dengan baik.
2. Ibu, Bapak di surga, keenam saudara, serta seluruh keluarga yang selalu memberikan segala dukungan utama kepada penulis.
3. Terimakasih untuk istriku tercinta Asti Dwi Irfianti di surga, yang telah mendorong saya dengan tidak bosannya untuk melanjutkan studi S2. AA senantiasa selalu mendoakanmu, insyaallah AA mau menjalankan amanahmu.
4. Bapak Waskitho Wibisono selaku pembimbing I yang telah banyak meluangkan waktu untuk dengan sangat sabar mendidik, membimbing, menjadi teman diskusi, serta mengajak penulis untuk berpikir lebih keras dari biasanya dalam menyelesaikan penelitian ini.
5. Bapak Ary Mazaharudin Shiddiqi selaku pembimbing II yang telah banyak meluangkan waktu untuk diskusi.
6. Seluruh staf dosen, staf tata usaha dan karyawan perpustakaan Jurusan Teknik Informatika, Institut Teknologi Sepuluh Nopember yang telah memfasilitasi penulis sejak awal kuliah.

7. Seluruh rekan-rekan S2 Informatika atas bantuan dan diskusi selama penelitian.
8. Teman materikulasi dan Teman sebimbangan yang selalu mengajarkan saya banyak hal, mas Riko Anshori, mas Awwib Maulana, mas Agung yang selalu memberikan semangat.
9. Semua pihak yang tidak dapat dituliskan satu per satu oleh penulis, terima kasih banyak atas doa dan dukungannya.
10. Andikawati Fitriasaki terimakasih supportnya, yang telah membantu koreksian selama penyusunan penelitian ini.
11. A Maman Suherman yang selalu support untuk kuliahku sampai bisa menyelesaikan penelitianku ini.

Semoga Allah SWT senantiasa menyayangi, menguatkan, memampukan, dan menunjukkan jalan yang terbaik atas semua kebaikan yang telah diberikan. Penulis menyadari bahwa laporan penelitian ini tentunya masih jauh dari kesempurnaan. Oleh sebab itu, saran dan kritik sangat diharapkan untuk perbaikan dimasa yang akan datang. Semoga laporan penelitian ini dapat bermanfaat bagi penulis dan pembaca pada umumnya.

Surabaya, Januari 2020



Ade Hendi

## DAFTAR ISI

<b>ABSTRAK</b> .....	i
<b>ABSTRACT</b> .....	iii
<b>KATA PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR TABEL</b> .....	xi
<b>DAFTAR GAMBAR</b> .....	xiii
<b>BAB I PENDAHULUAN</b> .....	1
1. 1. Latar Belakang.....	1
1. 2. Rumusan Masalah.....	3
1. 3. Batasan Masalah.....	3
1. 4. Tujuan Penelitian.....	3
1. 5. Manfaat Penelitian .....	4
1. 6. Kontribusi Penelitian .....	4
<b>BAB 2 KAJIAN PUSTAKA</b> .....	5
2. 1. Konsumsi Energi pada Perangkat Bergerak .....	5
2. 2. <i>Context-Aware System</i> .....	6
2. 3. <i>Indoor Positioning System</i> .....	6
2. 4. WIFI.....	7
2. 5. <i>Bluetooth</i> .....	8
2. 6. <i>Bandwith</i> .....	8
2. 7. ZigBee Radio.....	9
2. 8. <i>Accelerometer</i> .....	10
2. 9. <i>Gyroscope</i> .....	11
2. 10. <i>Magnetometer</i> .....	11
2. 11. <i>Point of Fingerprints</i> .....	11
2. 12. <i>K-Nearest Neighbors Algorithm (KNN)</i> .....	12
2. 13. <i>Euclidean Distance of Signal Strength</i> .....	12
2. 14. <i>Cluster Filtered KNN (CFK)</i> .....	13
2. 15. <i>Change Detection</i> .....	14
<b>BAB 3 METODOLOGI PENELITIAN</b> .....	17

3. 1.	Perumusan Masalah .....	17
3. 2.	Studi Literatur.....	18
3. 2.1.	<i>Fingerprint</i> .....	18
3. 2.2.	<i>Received Signal Strength Indicator</i> .....	20
3. 2.3.	Adaptive Sampling .....	21
3. 3.	Desain Sistem .....	22
3. 4.	Desain Algoritma dan Implementasi .....	24
3. 4.1.	Deteksi Pergerakan <i>User</i> Menggunakan <i>Bluetooth</i> .....	26
3. 4.2.	Deklarasi Perijinan Akses Komponen .....	27
3. 4.2.1.	Proses WiFi dan <i>Scanning</i> .....	27
3. 4.2.2.	Proses <i>Sampling Fingerprint</i> .....	29
3. 4.2.3.	Proses Deteksi <i>Bluetooth</i> .....	30
3. 4.3.	Prediksi <i>Change Detection</i> .....	31
3. 5.	Pengujian.....	33
3. 5.1.	Prediksi <i>Motion Change Detection</i> .....	34
3. 5.2.	Variasi <i>S Sliding Window</i> terhadap Akurasi <i>Prediksi Motion Change Detection</i> .....	35
3. 5.3.	<i>Entropy based Change Detection</i> .....	36
3. 5.4.	<i>Kullback-Leibler based Change Detection</i> .....	36
3. 5.5.	<i>Cosine Distance based Change Detection</i> .....	37
3. 5.6.	<i>Silent Zone</i> .....	37
3. 5.6.	Scanning Sinyal WiFi .....	40
3. 5.7.	WiFi <i>Fingerprint</i> .....	43
3. 5.8.	Estimasi Posisi.....	43
3. 5.9.	Efisiensi Energi Baterai .....	43
3. 5.10.	Presentasi Posisi .....	44
3.5.11.	Evaluasi.....	44
3. 5.	Jadwal Penelitian .....	45
<b>BAB 4 HASIL DAN PEMBAHASAN .....</b>		<b>47</b>
4. 1.	Tahapan Penelitian.....	47
4. 2.	Langkah-langkah Uji Coba .....	48
4.2.1.	<i>Skenario Pengujian</i> .....	48
4.2.1.1.	Skenario Pengujian Performa .....	48
4.2.1.2.	Skenario Pengujian Fungsionalitas .....	49
4. 3.	Hasil Uji Coba dan Analisis .....	50

4. 3.1. Hasil Uji Coba Performa .....	50
4. 3.2. Hasil Uji Coba Fungsionalitas .....	77
<b>BAB 5 KESIMPULAN DAN SARAN .....</b>	<b>81</b>
5.1. Kesimpulan.....	81
5.2. Saran .....	83
<b>DAFTAR PUSTAKA .....</b>	<b>85</b>

*[Halaman ini sengaja dikosongkan]*

## DAFTAR TABEL

Tabel 2.1 Algoritma <i>Cluster Filtered KNN</i> .....	13
Tabel 2.2 Algoritma <i>Hierarchical Clustering</i> .....	14
Tabel 3.1 Spesifikasi Bluetooth.....	27
Tabel 3.2 Skenario Pengujian .....	35
Tabel 3.3 Ringkasan analisis probabilitas kumulatif.....	40
Tabel 3.4 Jadwal Kegiatan Penelitian.....	45
Tabel 4.1 Ringkasan hasil penggunaan daya baterai .....	60
Tabel 4.2 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>Bluetooth</i> dengan 10 data training dengan nilai <i>Threshold</i> 11 .....	60
Tabel 4.3 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>Bluetooth</i> dengan 10 data training dengan nilai <i>Threshold</i> 8.....	61
Tabel 4.4 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>Bluetooth</i> dengan 10 data training dengan nilai <i>Threshold</i> 6.....	61
Tabel 4.5 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>Bluetooth</i> dengan 10 data training dengan nilai <i>Threshold</i> 10.....	62
Tabel 4.6 Ringkasan akurasi <i>change detection</i> dengan Bluetooth .....	62
Tabel 4.7 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>Accelerometer</i> dengan 10 data training dengan nilai <i>Threshold</i> 1,238.....	63
Tabel 4.8 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>accelerometer</i> dengan 10 data training dengan nilai <i>Threshold</i> 0,818 .....	63
Tabel 4.9 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>accelerometer</i> dengan 10 data training dengan nilai <i>Threshold</i> 0,519 .....	64
Tabel 4.10 Hasil uji coba akurasi <i>change detection</i> menggunakan <i>accelerometer</i> dengan 10 data training dengan nilai <i>Threshold</i> 1,117 .....	64
Tabel 4.11 Ringkasan akurasi <i>change detection</i> dengan <i>accelerometer</i> .....	65
Tabel 4.12 Ringkasan pengaruh akurasi <i>change detection</i> dengan <i>Bluetooth</i> terhadap penghematan baterai .....	67
Tabel 4.13 Ringkasan pengaruh akurasi <i>change detection</i> dengan <i>accelerometer</i> terhadap penghematan baterai .....	70

Tabel 4.14 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 3500$ ) dengan IPS <i>scanning</i> tanpa <i>change detection</i> .....	71
Tabel 4.15 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>Bluetooth threshold</i> 11 .....	72
Tabel 4.16 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>Bluetooth threshold</i> 8 .....	72
Tabel 4.17 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>Bluetooth threshold</i> 6 .....	73
Tabel 4.18 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>Bluetooth threshold</i> 10 .....	73
Tabel 4.19 Ringkasan akurasi posisi dengan <i>change detection</i> menggunakan <i>Bluetooth</i> .....	74
Tabel 4.20 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>accelerometer threshold</i> 1,238 .....	74
Tabel 4.21 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>accelerometer threshold</i> 0,818 .....	75
Tabel 4.22 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>accelerometer threshold</i> 0,519 .....	76
Tabel 4.23 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan <i>change detection</i> menggunakan <i>accelerometer threshold</i> 1,117 .....	76
Tabel 4.24 Ringkasan akurasi posisi dengan <i>change detection</i> menggunakan <i>accelerometer</i> .....	77



Tabel 4.16 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik  
POF ID 10 ( $x = 350, y = 350$ ) IPS dengan *change detection* menggunakan  
*Bluetooth threshold* 8..... 72

Tabel 4.16 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik  
POF ID 10 ( $x = 350, y = 350$ ) IPS dengan *change detection* menggunakan  
*Bluetooth threshold* 8..... 72

*[Halaman ini sengaja dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Alur Kerja Accelerometer .....	10
Gambar 3.1 Penentuan Posisi <i>Fingerprint</i> .....	19
Gambar 3.2 Kekuatan Sinyal.....	21
Gambar 3.3 Desain Sistem.....	23
Gambar 3.4 Trasnmisi Sinyal <i>Bluetooth</i> .....	24
Gambar 3.5 Alur Kerja Sistem .....	25
Gambar 3.6 Deklarasi Perijinan Akses Komponen .....	27
Gambar 3.7 Alur Kerja Proses Prediksi <i>motion change detection</i> .....	33
Gambar 3.8 Penempatan Perangkat WiFi .....	34
Gambar 3.9 Proses <i>Sliding Windows</i> .....	36
Gambar 3.10 <i>Raw Data</i> Pada Bluetooth .....	37
Gambar 3.11 <i>Raw Data</i> Pada <i>accelerometer</i> .....	37
Gambar 3.12 Selisih <i>Absolute</i> Hasil Training <i>Raw Data</i> Pada <i>Bluetooth</i> .....	38
Gambar 3.13 Selisih <i>Absolute</i> Hasil Training <i>Raw Data</i> Pada <i>accelerometer</i> .....	38
Gambar 3.14 Grafik Persebaran Probabilitas Kumulatif .....	39
Gambar 3.15 <i>Map</i> Titik Lokasi .....	41
Gambar 3.16 <i>Login User</i> .....	41
Gambar 3.17 <i>Raw Data WiFi Scanning</i> .....	42
Gambar 3.18 <i>Scanning WiFi</i> .....	42
Gambar 4.1 Penurunan Daya Baterai IPS <i>Scanning</i> .....	51
Gambar 4.2 Penurunan daya baterai kondisi diam 100% dengan <i>Bluetooth</i> .....	51
Gambar 4.3 Penurunan daya baterai kondisi diam 100% dengan <i>Accelerometer</i> ..	52
Gambar 4.4 Perbandingan penurunan daya baterai kondisi diam 100% dengan IPS periodik .....	52
Gambar 4.5 Penurunan daya baterai kondisi gerak 10% dengan <i>Bluetooth</i> .....	53
Gambar 4.6 Penurunan daya baterai kondisi gerak 10% dengan <i>Accelerometer</i> ..	53
Gambar 4.7 Perbandingan penurunan daya baterai kondisi gerak 10% dengan IPS periodik .....	53
Gambar 4.8 Penurunan daya baterai kondisi gerak 30% dengan <i>Bluetooth</i> .....	54
Gambar 4.9 Penurunan daya baterai kondisi gerak 30% dengan <i>Accelerometer</i> ..	54

Gambar 4.10 Perbandingan penurunan daya baterai kondisi gerak 30% dengan IPS periodik .....	55
Gambar 4.11 Penurunan daya baterai kondisi gerak 50% dengan <i>Bluetooth</i> .....	55
Gambar 4.12 Penurunan daya baterai kondisi gerak 50% dengan <i>Accelerometer</i>	56
Gambar 4.13 Perbandingan penurunan daya baterai kondisi gerak 50% dengan IPS periodik .....	56
Gambar 4.14 Penurunan daya baterai kondisi gerak 75% dengan <i>Bluetooth</i> .....	57
Gambar 4.15 Penurunan daya baterai kondisi gerak 75% dengan <i>Accelerometer</i>	57
Gambar 4.16 Perbandingan penurunan daya baterai kondisi gerak 75% dengan IPS periodik .....	57
Gambar 4.17 Penurunan daya baterai kondisi gerak 100% dengan <i>Bluetooth</i> .....	58
Gambar 4.18 Penurunan daya baterai kondisi gerak 100% dengan <i>Accelerometer</i> .....	58
Gambar 4.19 Perbandingan penurunan daya baterai kondisi gerak 100% dengan IPS periodik .....	59
Gambar 4.20 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>Bluetooth threshold 11</i> .....	65
Gambar 4.21 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>Bluetooth threshold 8</i> .....	66
Gambar 4.22 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>Bluetooth threshold 6</i> .....	66
Gambar 4.23 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>Bluetooth threshold 10</i> .....	66
Gambar 4.24 Perbandingan penurunan daya baterai dengan <i>threshold</i> pada <i>change detection</i> menggunakan <i>Bluetooth</i> .....	67
Gambar 4.25 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>accelerometer threshold 1,238</i> .....	68
Gambar 4.26 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>accelerometer threshold 0,818</i> .....	68
Gambar 4.27 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>accelerometer threshold 0,519</i> .....	69

Gambar 4.28 Penurunan daya baterai dengan <i>change detection</i> menggunakan <i>accelerometer threshold</i> 1,117 .....	69
Gambar 4.29 Perbandingan penurunan daya baterai dengan <i>threshold</i> pada <i>change detection</i> menggunakan <i>accelerometer</i> .....	69
Gambar 4.30 <i>Scanning</i> sinyal Bluetooth.....	71
Gambar 4.31 Respon Sensor <i>Accelerometer</i> .....	73
Gambar 4.32 <i>Scanning</i> Sinyal WiFi .....	74
Gambar 4.33 Hasil WiFi <i>Scanning</i> .....	74

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PSEUDOCODE

Pseudocode 3.1 Inisialisasi WiFi dan Scanning .....	28
Pseudocode3.2 Sampling Fingerprint .....	29
Pseudocode3.3 Proses Deteksi <i>Bluetooth</i> .....	30
Pseudocode3.4 <i>Change Detection</i> .....	32

# BAB 1

## PENDAHULUAN

### 1.1. Latar Belakang

Perkembangan terbaru dalam teknologi komunikasi mengalami peningkatan yang sangat signifikan dari waktu ke waktu dan menjadi salah satu aspek yang tergolong penting. Aspek utama teknologi komunikasi adalah kemampuan mengumpulkan data dari dunia sekitar hingga memproses untuk dapat digunakan pada berbagai keperluan. Meningkatnya mobilitas dan kebutuhan manusia terhadap teknologi komunikasi menyebabkan perkembangan teknologi nirkabel menjadi sangat populer. Disamping penggunaan alat yang praktis, seperti tidak memerlukan adanya penggunaan kabel-kabel, teknologi nirkabel dapat diprogram dan dilengkapi dengan satu set sensor yang tertanam didalamnya, antara lain penerima GPS, *accelerometer*, *gyroscope* dan *magnetometer*. Selain itu, layanan yang diberikan semakin beraneka ragam, fitur-fitur yang semakin canggih membuat teknologi nirkabel lebih diunggulkan.

*Global Positioning System* (GPS) merupakan sensor yang sangat dominan untuk layanan informasi pelacakan suatu objek atau orang yang berada diluar ruangan. Sedangkan Pemakaian GPS dalam suatu ruangan memiliki kendala antara lain ketersediaan jalur *Line-of-Sight* (Wasiq Waqar et al., 2016). Ketersediaan jalur LOS (*Line-of-Sight*) yang rendah menjadi kendala dalam menemukan antara pemancar dan penerima sinyal, hilangnya LOS diantara perangkat seluler dan satelit dapat membuat kinerja GPS kurang maksimal di dalam ruangan (Xuan Du., 2018). Layanan informasi untuk pelacakan diluar ruangan sangat populer, disebabkan karena banyak *smartphone* yang memiliki GPS, tetapi perangkat GPS tidak cocok untuk pelacakan dalam ruangan (Kyohei Suginoet al., 2016). Pada penelitian sebelumnya menyebutkan bahwa orang menghabiskan 10-20% dari waktu mereka diluar ruangan dan lebih dari 70% menunjukan aktivitas berasal dalam ruangan. Hal ini menunjukan potensi besar untuk layanan informasi pelacakan dalam ruangan seperti *Indoor Positioning System* (IPS) (Wasiq Waqar et al., 2016).



*Indoor Positioning System* merupakan sebuah layanan informasi yang menggunakan teknologi nirkabel untuk menemukan objek atau seseorang di dalam gedung. *Indoor Positioning System* telah berhasil digunakan pada beberapa aplikasi pelacakan dan manajemen persediaan (Hui Liu et al., 2007). Permintaan untuk layanan berbasis *Indoor Positioning System* meningkat secara signifikan. Aplikasi yang menggunakan layanan *Indoor Positioning System* (IPS) diantaranya sistem navigasi untuk melacak orang-orang di Rumah Sakit, monitoring pegawai di Rumah Sakit, membimbing orang buta, pelacakan anak kecil atau orang Lanjut usia, sistem navigasi belanja di mall (Jianwei Niu et al., 2015).

Beberapa teknologi yang dapat digunakan untuk membangun layanan *Indoor Positioning System* (IPS) dengan penggunaan daya rendah, seperti Zigbee, WiFi dan *Bluetooth*. Pada penelitian ini, peneliti menggunakan metode *Wireless Ethernet IEEE 802.11* dan *accelerometer* sebagai objek penelitian. *Wireless Ethernet IEEE 802.11* dan *accelerometer*, merupakan teknologi nirkabel dan sensor yang dapat digunakan untuk layanan *Indoor Positioning System* (IPS). Disamping biayanya tergolong murah, pemakaiannya mudah untuk diakses (Gints Jēkabsons et al., 2011), (Zhe Xiang et al., 2004). *Wireless Ethernet IEEE 802.11* menggunakan *Received Signal Strength Indicator* (RSSI) yang merupakan infrastruktur untuk mengaktifkan *fingerprint* berdasarkan posisi (Artur Baniukevic et al., 2013).

Pada penelitian ini, mekanisme *change detection* dengan pengambilan sampel secara adaptif menggunakan *accelerometer* sebagai trigger diusulkan untuk meningkatkan efisiensi energi. *Change detection* merupakan proses penting sebagai analisa (Dorel Aiordachioaie et al., 2014) untuk mengukur rentang nilai pada *accelerometer*.

Masalah yang diangkat dalam penelitian ini adalah cara melakukan *sampling* kekuatan sinyal pada WiFi secara adaptif untuk penghematan energi dalam *Indoor positioning system* (IPS). Penelitian telah mengarah pada penciptaan suatu sistem yang sadar akan energi yang dirancang bertujuan mengurangi konsumsi energi pada aplikasi IPS (Hasan Sajid et al., 2018). Pada penelitian terkait yang lain juga meneliti *scanning* kekuatan sinyal secara adaptif, untuk tujuan penghematan energi dengan sistem pengenalan aktivitas yang

menggunakan sistem lokasi berbasis WiFi yang dikombinasikan dengan *accelerometer* untuk identifikasi gerak (Alvarez-Alvarez et al., 2013).

Teknologi *Indoor Positioning System* (IPS) merupakan perangkat bergerak yang mengkonsumsi cukup banyak daya baterai. Hal ini disebabkan karena proses *scanning* yang terus menerus membutuhkan daya yang cukup banyak sehingga mengurangi daya baterai secara *runtime* (sebagai contoh perangkat bergerak mengkonsumsi lebih dari 40% daya baterai yang keseluruhannya dipergunakan pada fasilitas WiFi (Jianwei Niu et al., 2015). Berdasarkan latar belakang diatas, penelitian ini mengambil judul “Pengembangan Mekanisme *Change Detection* Untuk Efisiensi Energi Pada *WiFi-Based Indoor Positioning System*”.

## **1.2. Rumusan Masalah**

Dari latar belakang yang telah dijelaskan diatas, dapat dirumuskan permasalahan sbb:

1. Bagaimana mekanisme *change detection* untuk *scanning sampling* terhadap kekuatan sinyal secara adaptif untuk penghematan energi pada *WiFi - Indoor Positioning System*.
2. Bagaimana pengaruh akurasi *change detection* terhadap mekanisme *WiFi - Indoor Positioning System* secara adaptif.

## **1.3. Batasan Masalah**

Berdasarkan uraian diatas, penulis membatasi permasalahan sbb:

1. Penelitian dilakukan dengan uji coba langsung dengan *real device*.
2. Penelitian ini berfokus pada metode penghematan daya pada proses *WiFi scanning*.
3. Penelitian menggunakan tiga WiFi dengan tiga ruangan yang dipisahkan oleh tembok atau dinding.

## **1.4. Tujuan Penelitian**

Berdasarkan perumusan masalah yang dijelaskan pada sub bab 1.2, penelitian ini bertujuan untuk :

1. Melakukan *scanning* untuk *sampling* kekuatan sinyal pada WiFi secara adaptif dengan *change detection* untuk penghematan energi.
2. Meneliti pengaruh akurasi *change detection* terhadap mekanisme WiFi *Indoor Positioning System* secara adaptif.

### **1.5. Manfaat Penelitian**

Penelitian diharapkan dapat mengurangi pemborosan konsumsi daya baterai pada proses WiFi *scanning* Sistem Pendeteksi Posisi dalam ruangan tanpa menghilangkan keakuratan informasi posisi secara signifikan.

### **1.6. Kontribusi Penelitian**

Perbaikan mekanisme penghematan daya dengan *change detection* pada proses WiFi *scanning* pada Sistem Pendeteksi Posisi dalam ruangan dengan mempertimbangkan ketersediaan daya tanpa mengurangi keakuratan informasi posisi secara signifikan.

## **BAB 2**

### **KAJIAN PUSTAKA**

Pada bab ini peneliti akan memberikan penjelasan studi literatur yang sebelumnya dilakukan pada topik terkait untuk tesis ini. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap objek penelitian yang sedang diteliti. Guna mendukung material riset, pada *literature view* dijelaskan pada uraian di bawah ini.

#### **2.1. Konsumsi Energi pada Perangkat Bergerak**

Konsumsi energi telah dan akan selalu menjadi salah satu masalah terbesar dalam jaringan nirkabel. Untuk memiliki mobilitas, perangkat bergerak mengandalkan daya baterai, dan di sisi lain, dibatasi oleh kapasitas baterai terlepas dari peningkatan dalam fungsionalitas dan *hardware infrastructure*. Memahami konsumsi energi pada sistem perangkat bergerak adalah sangat penting untuk melakukan penghematan daya lebih lanjut. Untuk mengungkap karakteristik nilai dari suatu energi dalam sistem perangkat bergerak, para peneliti telah melakukan studi pengukuran yang memberikan wawasan signifikan dari aspek yang berbeda. Pengukuran pada literatur (Mark Stemm et al., 2008) menunjukkan bahwa daya yang terkuras oleh antarmuka jaringan merupakan sebagian besar dari total daya yang digunakan oleh perangkat bergerak. Pada *Proceedings* (L. M. Feeney et al., 2001) juga dilakukan pengujian peluang optimalisasi terhadap aplikasi khusus untuk mengurangi konsumsi energi dari suatu antarmuka jaringan. Untuk memperoleh pengetahuan yang lebih praktis, dalam mengukur konsumsi daya antarmuka jaringan nirkabel WiFi IEEE 802.11 yang beroperasi di lingkungan jaringan ad hoc, dan menerapkan rumus persamaan linear untuk menggambarkan perilaku konsumsi energi (L. M. Feeney et al., 2001). Sebuah studi pada *Proceedings* (Niranjan Balasubramanian et al., 2009) membandingkan fitur konsumsi energi dari tiga teknologi jaringan perangkat bergerak sebagai berikut: 3G, GSM, dan WiFi. Temuan utama berikut adalah pada 3G, sebagian besar energi yang disebut sebagai *tail energy*, terbuang dalam kondisi daya yang tinggi setelah selesainya proses transfer data; GSM juga

menunjukkan tren yang sama dengan *tail time* yang jauh lebih pendek; WiFi, meskipun biaya sambungannya sebanding dengan *tail energy*, tapi memiliki transfer data yang jauh lebih efisien daripada 3G. Pada literatur Proceedings (A. Carroll et al., 2010) mengukur konsumsi daya perangkat bergerak modern, dipecah menjadi perangkat subsistem utama (CPU, memori, layar sentuh, perangkat keras grafis, audio, penyimpanan, dan berbagai antarmuka jaringan), di bawah berbagai skenario penggunaan secara realistis. Tuntutan energi pada platform perangkat bergerak juga telah ditandai dalam perspektif perilaku pengguna (Narseo Vallina-Rodriguez et al., 2011) yang dapat membantu untuk saling memahami ketergantungan antara sumber daya dalam konteks tertentu.

## **2.2. Context-Aware System**

*Context* merupakan sekumpulan informasi-informasi yang dapat digunakan untuk menggambarkan situasi dari suatu entitas. Entitas yang dimaksud dapat berupa orang, tempat, maupun benda yang dianggap sesuai untuk melakukan interaksi dengan *user* dan aplikasi. *Context* dapat mencakup informasi seperti lokasi *user*, *bandwidth* jaringan, aktivitas *user*, emosi *user* dan lain-lain. Maka, disimpulkan bahwa *Context-Aware System* merupakan sebuah sistem yang mampu menggunakan informasi dari *context* untuk memberikan informasi yang relevan kepada *user*, tergantung kepada aktivitas user itu sendiri (Seng Loke, 2009). Pada penelitian ini, sedang diteliti sebuah aplikasi *Context-Aware* yang memiliki kemampuan mengetahui posisi *user* di dalam ruangan dengan menggunakan kekuatan sinyal WiFi.

## **2.3. Indoor Positioning System (IPS)**

*Indoor Positioning System* merupakan salah satu sistem yang menerapkan konsep *Context-Aware*. Sistem ini dapat menemukan posisi suatu obyek di dalam ruangan, baik berupa orang, benda, dan lain-lain. GPS (Global Positioning System) merupakan sistem penentuan posisi yang paling populer, yang paling sering digunakan pada beberapa aplikasi saat ini. Namun untuk di dalam ruangan, GPS tidak dapat berfungsi semestinya seperti pada *outdoor*, hasil yang diperoleh tidak akurat karena keterbatasan yang dimiliki. Oleh karena itu, pada penelitian

ini peneliti akan meneliti sebuah aplikasi yang menggunakan WiFi dengan memanfaatkan kekuatan sinyal radio nya.

#### **2.4. WiFi**

WiFi merupakan salah satu perangkat *local area network* nirkabel yang umum dan paling banyak digunakan saat ini. Wifi diterapkan dalam *local network* dengan mengatur penyediaan perangkat *wireless Access Points* (AP) agar terhubung ke dalam suatu jaringan secara nirkabel dan menyediakan konektivitas pada kemungkinan mobilitas di sekitar perangkat pada *local area* sedang terhubung ke AP. Aktivitas ini menunjukkan bahwa dimana pun perangkat *local area* berada, dapat berkomunikasi dengan AP. AP biasanya merupakan jaringan yang terhubung ke Internet, yang menyediakan konektivitas kemungkinan berkomunikasi dengan perangkat pada berbagai *local area network*. WiFi muncul sebagai nama merek pemasaran *wireless local area network* (WLAN) terbaru, yang didasarkan pada *Institute of Electrical and Electronics Engineers* standar IEEE 802.11. IEEE 802.11 beroperasi di bawah protokol berbeda dengan beragam frekuensi: 2.4GHz, 3.6GHz, 5GHz dan 60GHz; semua menyediakan rentang dan kecepatan komunikasi yang berbeda. IEEE 802.11 pertama kali dikembangkan pada tahun 1997, kemudian menyediakan protokol tipe "a" pada tahun 1999, "b" pada tahun 1999, "g" pada tahun 2003, "n" pada tahun 2009 dan "Ac" pada 2013; mereka semua mendorong peluang lebih lanjut (Wi-Fi Alliance, 2015).

WiFi adalah sistem *half-duplex*, yang berarti mendukung komunikasi dari kedua arah tetapi tidak pada saat yang sama. Sistem ini bekerja dengan pengkodean data pada gelombang radio (pembawa), yang kemudian diterapkan ke suatu frekuensi dengan menyiarkannya. Gelombang radio kemudian diterima dan diterjemahkan kembali menjadi data yang dapat dibaca. Untuk mengirim data pada suatu frekuensi radio, salah satu dari dua cara modulasi sinyal radio harus diterapkan, baik dengan *amplitude modulation* (AM) atau dengan *frequency modulation* (FM).

Dalam beberapa tahun terakhir, jaringan *local area* nirkabel (WLAN) telah digunakan di lingkungan dalam ruangan terutama area publik, seperti universitas, rumah sakit, bandara dan lain-lain. Saat ini WiFi (IEEE 802.11) telah menjadi

teknologi representatif untuk WLAN. Sebagai tambahan untuk transmisi data broadband, jaringan WiFi juga menyediakan alternatif yang baik untuk pemosisian dalam ruangan.

### **2.5. Bluetooth**

*Bluetooth* adalah teknologi komunikasi nirkabel yang banyak digunakan dan telah menjadi sangat populer di rumah seperti, pengeras suara nirkabel menggunakan teknologi *Bluetooth* untuk menerima musik dari perangkat. Teknologi *Bluetooth* dirancang untuk komunikasi jarak pendek dan aplikasi utamanya adalah untuk bertindak sebagai protokol penggantian kabel pada *keyboard* nirkabel, *headset* nirkabel dll. Jaraknya rendah dan daya rendah merupakan karakteristik yang membuat *Bluetooth* sangat cocok dalam komunikasi antara perangkat kecil nirkabel.

*Bluetooth* merupakan perangkat protokol yang dapat digunakan sebagai perangkat untuk berkomunikasi, dan merupakan teknologi yang beroperasi pada frekuensi 2,4 GHz. Frekuensi ini dapat disebut sebagai *Industrial, Scientific and medical* (ISM) dan digunakan oleh beberapa teknologi lainnya. Untuk meminimalkan gangguan, *Bluetooth* menggunakan teknik yang disebut sebagai *normal frequency hopping*. *Normal frequency hopping* adalah teknik untuk menghindari gangguan dengan mengganti pembawa di antara beberapa frekuensi saluran dengan kecepatan tinggi secara *random* (Bluetooth, 2015). *Bluetooth* disebut juga sebagai teknologi nirkabel jarak pendek (biasanya 10-15 m) yang dikembangkan untuk menyediakan skenario komunikasi berbiaya rendah dan *bandwidth* rendah (Yapeng Wang et al., 2013).

### **2.6. Bandwidth**

*Bandwidth* atau lebar pita dalam istilah komputer merupakan kuota atau lebar cakupan frekuensi yang dipergunakan oleh sinyal dalam medium transmisi elektronik (Seputar Tekno, 2013). Pengertian lainnya, *Bandwidth* merupakan satuan ukuran lalu lintas data yang menunjukkan jumlah data yang ditransfer dalam satuan waktu tertentu (Jon Raasch, 2010).

Pengertian *Bandwidth* pada *web hosting* sedikit berbeda. *web hosting bandwidth* adalah nilai maksimum besaran transfer data (gambar, video, suara dan lainnya) yang terjadi antara *server hosting* dengan klien dalam suatu periode tertentu (DZul Fadli, 2012) dan berpengaruh pada kinerja aliran paket data di lalu lintas jaringan. Semakin besar lalu lintas yang terjadi pada sebuah *web hosting* semakin besar pula *bandwidth* yang diperlukan. Ketika *bandwidth* yang digunakan untuk lalu lintas data melebihi kapasitas, maka data tersebut tidak dapat dialirkan pada suatu jaringan.

Pengukuran *bandwidth* dapat dilakukan dengan mengukur jumlah data yang ditransmisikan kemudian dibagi dengan waktu yang dibutuhkan dari jumlah yang ditransmisikan hingga data diterima. Rumus pengukuran *bandwidth* ditunjukkan sebagai berikut :

$$B = \frac{\text{Ukuran data}}{\text{Periode waktu transfer}}$$

Keterangan:

B : *bandwidth*

## 2.7. ZigBee Radio

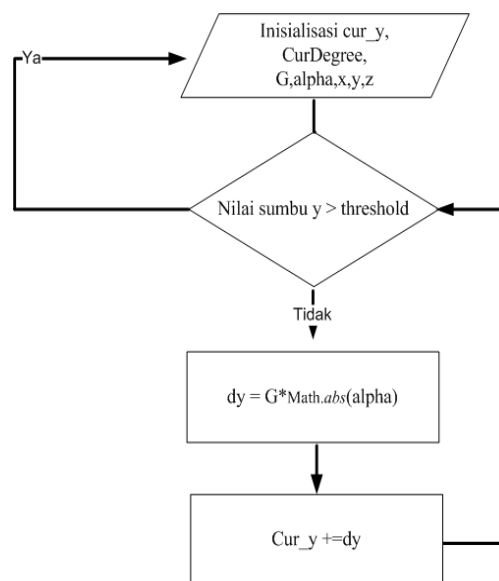
ZigBee adalah standar jaringan nirkabel industri yang paling populer untuk menghubungkan sensor, instrumentasi dan sistem kontrol. ZigBee, merupakan spesifikasi perangkat untuk komunikasi dalam *wireless personal area network* (WPAN), telah disebut "*Internet of things*". ZigBee adalah *open protocol, global, packet-based protocol* yang dirancang untuk menyediakan arsitektur yang mudah digunakan untuk jaringan nirkabel berdaya rendah yang aman dan andal. ZigBee dan IEEE 802.15.4 adalah standar jaringan nirkabel dengan kecepatan data rendah, yang dapat menggantikan kabel yang rentan dan mahal pada aplikasi kontrol industri. Peralatan kontrol proses dapat ditempatkan di mana saja dan masih dapat berkomunikasi dengan sistem lainnya. Di samping semua itu juga dapat dipindahkan, karena jaringan tidak mengenal tentang lokasi fisik dari sebuah sensor, pompa atau katup.



Standar ZigBee RF4CE meningkatkan standar IEEE 802.15.4 dengan menyediakan lapisan jaringan sederhana dan profil aplikasi standar yang dapat digunakan untuk membuat solusi pada perangkat elektronik *multi-vendor* yang dapat dioperasikan. Manfaat teknologi ini sangat banyak, beberapa aplikasi ZigBee meliputi: Otomatisasi rumah dan kantor, Industri Otomasi, Pemantauan medis, Sensor berdaya rendah, Kontrol HVAC, dan banyak kontrol serta penggunaan pemantauan lainnya.

### 3.8. Accelerometer

*Accelerometer* merupakan sensor yang digunakan untuk mendeteksi adanya getaran dan pergerakan pada perangkat bergerak (*smartphone*) baik statis maupun dinamis (G. Millette, 2012). Pada perangkat bergerak (*smartphone*) terdiri dari tiga sumbu x, y, dan z dengan orientasi yang berbeda-beda setiap sumbu. Dalam penentuan posisi menggunakan *smartphone*, nilai yang dihasilkan dari sensor ini masih dipengaruhi oleh gaya gravitasi, untuk menghilangkannya bisa menggunakan metode *low-pass filter*. Hasil dari *accelerometer* ini biasanya digunakan untuk *step detection* yang diolah untuk memperkirakan pergerakan dari *user*. Alur kerja dari *accelerometer* dapat dilihat pada Gambar 2.1.



Gambar 2.1. Alur Kerja *Accelerometer*

Keterangan gambar di atas menunjukkan bahwa untuk mengetahui apakah *user* sedang bergerak atau tidak. Untuk mengetahui pergerakan tersebut yang diperhatikan cukup nilai dari sumbu *y* apakah melebihi *threshold* yang ditentukan atau tidak. Jika nilai sumbu *y* melebihi *threshold*, maka *user* dianggap sedang bergerak. Keterangan dari variabel pada gambar di atas sebagai berikut: *dy* : Gravitasi Resultan, *G* : Gravitasi Bumi, *Math.abs* (*alpha*) : Sudut Kemiringan.

## 2.9. Gyroscope

*Gyroscope* adalah alat untuk mengukur orientasi. Fenomena yang disebut kunci gimbals adalah salah satu masalah utama dengan *gyroscope* mekanik. Ini terjadi karena dua dari tiga cincin gimbals disejajarkan di bidang yang sama karena rotasi. Hal ini mengurangi tingkat kebebasan sistem dan gimbals tidak lagi dapat berputar dan mempertahankan orientasinya. Dalam beberapa tahun terakhir, *gyroscope* murah yang diproduksi dengan teknologi MEMS telah tersedia secara luas. Sensor-sensor ini bekerja dengan cara yang mirip dengan *accelerometer* linier, karena mereka memberikan pembacaan *instant* dari kecepatan sudut. Nilai ini dapat direkam dan diintegrasikan dengan waktu untuk menghitung orientasi objek (Wasiq Waqar et al., 2016).

## 2.10. Magnetometer

*Magnetometer* adalah instrumen yang digunakan untuk mengukur kekuatan dan / atau arah medan magnet di sekitar instrumen. *Magnetometer* dapat dibagi menjadi dua tipe dasar: *magnetometer* skalar yang mengukur kekuatan total medan magnet yang menjadi sasarannya, dan *magnetometer* vektor yang memiliki kemampuan untuk mengukur komponen medan magnet dalam arah tertentu, relatif terhadap orientasi spasial perangkat (Wikipedia, 2011).

## 2.11. Point of Fingerprints (POF)

*Point of fingerprints* merupakan suatu titik yang berisi sekumpulan penanda atau sidik jari (*fingerprint*) hasil dari *sampling* yang telah dilakukan sebelumnya pada tiap ruangan (Jun MA et al., 2008). *Point of fingerprints* ini berupa tipe data tuple pada Persamaan

$$(q_i, q_{ii})_{i = 1,2,3, \dots, n} \quad j = 1,2,3, \dots, m \quad (2.1)$$

Variabel  $q$  merupakan koordinat *Point of Fingerprints* yang dilakukan *sampling*  $q_i = (x_i, y_i)$ ,  $x$  merupakan posisi sumbu  $x$  dan  $y$  merupakan posisi sumbu  $y$ . Sedangkan  $n$  adalah jumlah *point of fingerprints* dan  $m$  adalah jumlah *fingerprint* pada setiap *point of fingerprints*. Variabel  $r_{ij}$  pada Persamaan 2.2 merupakan *fingerprint* atau sidik jari yang berupa data kekuatan sinyal Wi-Fi yang telah ditentukan sebelumnya dalam satuan - dbm. Konstanta  $k$  adalah jumlah Wi-Fi yang digunakan.

$$r_i = (r_{ij1}, r_{ij2}, r_{ij3}, \dots, r_{ijk}) \quad (2.2)$$

### 2.12. *K-Nearest Neighbors Algorithm (KNN)*

Algoritma *K-Nearest Neighbor* merupakan salah satu metode yang digunakan untuk melakukan klasifikasi terhadap obyek berdasarkan data *training* yang memiliki jarak terdekat dari obyek itu. Data *training* diproyeksikan ke ruang berdimensi banyak. Setiap ruang dibagi menjadi bagian-bagian berdasarkan klasifikasi data *training*. Sebuah titik pada ruang ini termasuk dalam suatu kelas  $x$ , bila kelas  $x$  tersebut klasifikasi yang paling banyak ditemui pada  $k$  buah tetangga terdekat titik tersebut. Untuk menghitung jaraknya digunakan rumus *euclidean distance*. Nilai  $k$  yang terbaik untuk algoritma ini tergantung pada data. Pada umumnya nilai  $k$  yang tinggi akan mengurangi efek *noise* pada klasifikasi, tetapi membuat batasan antara setiap klasifikasi menjadi lebih kabur. Pemilihan nilai  $k$  yang sesuai akan mempengaruhi ketepatan klasifikasi data.

### 2.13. *Euclidean Distance of Signal Strength*

*Euclidean distance* digunakan untuk mencari jarak antara titik lokasi *user* saat itu dengan titik tetangga yang berada di sekitarnya. Pada Persamaan 2.3, perhitungan dilakukan menggunakan kekuatan sinyal Wi-Fi (Jun MA et al., 2008).

$$Dist(RSSV_{ij}, RSSV_{ij}) = \sqrt{\sum_{k=1}^n (RSS_{ijk} - RSS_{ljk})^2} \quad (2.3)$$

Dengan mencari jarak antara data kekuatan sinyal yang ditangkap posisi user saat itu ( $RSSV_i$ ) dengan data kekuatan sinyal setiap *point of fingerprints* ( $RSSV_j$ ) yang berada di sekitarnya dan  $i$  merupakan jumlah dari

*point of fingerprints*. RSSV merupakan sebuah vektor yang berisi *fingerprint*,  $j$  merupakan jumlah dari *fingerprint*, dan  $n$  adalah jumlah dari WiFi yang digunakan.

#### 2.14. Cluster Filtered KNN (CFK)

Algoritma *Cluster Filtered K-Nearest Neighbors* memperoleh tetangga  $k$  sama halnya dengan algoritma KNN pada umumnya (Jun MA et al., 2008). Dijelaskan pada Tabel 2.1, tidak seperti KNN yang mengambil semua tetangga terdekatnya dan menggunakan sistem voting, *Cluster Filtered K-Nearest Neighbors* melakukan *filtering* untuk menyeleksi tetangga mana saja yang berhak untuk dipilih. Awalnya dengan mencari nilai *center of point* (*average*) dari setiap tetangga, kemudian menggunakan pengelompokan (sesuai dengan lokasi fisik setiap tetangga) untuk mempartisi ke dalam beberapa kelompok dan hanya satu *cluster* yang akan terpilih.

**Tabel 2.1** Algoritma *Cluster Filtered KNN*

1.	Mencari <i>point of fingerprints</i> terdekat sejumlah $k$ dengan menggunakan algoritma <i>K-Nearest Neighbors</i> .
2.	Mencari nilai <i>center of cluster</i> dari setiap <i>point of fingerprints</i> yang dihasilkan.
3.	Menggunakan <i>clustering alghoritm</i> dengan membagi menjadi beberapa <i>cluster</i> .
4.	Pilih satu <i>cluster</i> sesuai dengan <b>CFKRule</b> .
5.	Ambil nilai rata-rata dari koordinat semua anggota pada <i>cluster</i> yang terpilih untuk menemukan lokasi <i>user</i>
	<b>CFK Rule:</b> <ol style="list-style-type: none"> <li>1. Pilihlah <i>cluster</i> yang memiliki anggota terbanyak.</li> <li>2. Pilihlah <i>cluster</i> yang memiliki <i>average RSSVdistance</i> terkecil apabila dua <i>cluster</i> memiliki anggota yang sama.</li> </ol>

Ada banyak metode clustering yang dapat digunakan untuk menyelesaikan permasalahan Indoor Positioning System. Pada penelitian ini, penulis menyelesaikan permasalahan IPS menggunakan hierarchical clustering (Jun MA et al., 2008). Proses clustering dengan metode hierarchical clustering cenderung sederhana. Awalnya setiap data ditempatkan pada setiap kelompok. Sepasang kelompok yang memiliki jarak terdekat kemudian

dijadikan dalam satu kelompok dan ditentukan koordinat baru dengan mencari rata-rata koordinatnya. Berlanjut terus-menerus sampai ke semua data. Untuk lebih lengkapnya, dapat dilihat pada Tabel 2.2.

**Tabel 2.2** Algoritma *Hierarchical Clustering*

1.	Tempatkan setiap obyek yang ada menjadi satu <i>cluster</i> .
2.	Hitung jarak antar <i>cluster</i> .
3.	Gabungkan dua <i>cluster</i> menjadi satu <i>cluster</i> baru yang memiliki jarak terpendek.
4.	Hitung kembali jarak dengan <i>cluster</i> yang baru.
5.	Jika jarak minimum lebih besar dari <b>threshold</b> , maka hentikan. Jika tidak lakukan kembali langkah 2.

### 2.15. *Change Detection*

Fokus penelitian ini adalah penghematan pada daya baterai dimana proses *scanning* kekuatan sinyal WiFi akan diproses sesuai dengan ada tidaknya perubahan. Pengurangan frekuensi *scanning* bertujuan untuk mengurangi konsumsi daya baterai. Prediksi ada atau tidaknya perubahan aktivitas pada posisi yang terjadi pada *user* dilihat dari ada atau tidaknya perubahan nilai RSSI yang dihasilkan dari *Bluetooth*. Secara umum, perubahan dikenal sebagai perpindahan dari satu kondisi ke kondisi lainnya (Shen-Shyang Ho et al., 2010). Pemilihan model yang tepat untuk proses identifikasi dan perkiraan terhadap parameter, digunakan untuk mendeteksi dengan menggunakan nilai-nilai kriteria merupakan suatu proses yang menyiratkan suatu perubahan berdasarkan nilai varian (Dorel Aiordachioaie et al., 2014). Deteksi perubahan dalam aliran data bergantung pada perkiraan dari probabilitas, bahwa data dalam dua *window* berurutan berasal dari distribusi yang berbeda. Memilih kriteria merupakan suatu hal yang perlu diutamakan dalam merancang prosedur deteksi perubahan (Ludmila I Kuncheva, 2013). Ada beberapa *change detection* terkait untuk mendeteksi perubahan di mana beberapa dari mereka mendeteksi perubahan dari pengamatan berasal dari distribusi stasioner. Ketika data mengalir dari waktu ke waktu, dan setidaknya untuk periode waktu yang besar, tidak mungkin mengasumsikan bahwa pengamatan itu dihasilkan secara acak sesuai dengan distribusi probabilitas stasioner. Setidaknya dalam sistem yang kompleks dan untuk periode waktu yang

besar, kita harus mengharapkan perubahan dalam distribusi data (Raquel Sebastião et al., 2007). Beberapa cara *change detection* untuk mendeteksi perubahan tersebut adalah : *Entropy based Change Detection*, *Kullback-Leibler based Change Detection* dan *Cosine Distance based Change Detection*.

Menurut (Duc-tan et al., 2015) perubahan dapat dideteksi dengan menggunakan alur *silent - movement mode*. Pada alur *silent - movement mode* terdapat sebuah rentang nilai yang disebut *silent zone*, yang mana rentang nilai tersebut merupakan rentang toleransi atas *noise* pada data.

*[Halaman ini sengaja dikosongkan]*

## **BAB 3**

### **METODOLOGI PENELITIAN**

Langkah-langkah penelitian yang akan dilakukan pada penelitian ini adalah sebagai berikut :

1. Perumusan masalah
2. Studi Literatur
3. Desain Sistem
4. Desain Algoritma dan Implementasi
5. Pengujian dan Evaluasi

#### **3.1. Perumusan Masalah**

Perumusan masalah yang diangkat dalam penelitian ini adalah cara melakukan *sampling* kekuatan sinyal pada WiFi secara adaptif dengan *change detection* untuk penghematan energi dalam *Indoor positioning system* (IPS). Dalam dunia seluler, masa pakai baterai menjadi perhatian besar. Pada saat yang sama, kemampuan perangkat bergerak untuk memberikan informasi lokasi dari *user* merupakan salah satu keunggulan pada fiturnya. Cara sebelumnya memberikan informasi lokasi dari *user* adalah melalui penggunaan GPS. Meskipun akurasi telah terbukti, tetapi terjadi pemborosan dalam hal masa pakai baterai. Karena itu, alternatif lain yang paling umum untuk penggunaan selain GPS adalah WiFi. Berkaitan dengan cara pengambilan sampel adalah mungkin salah satu pendekatan untuk manajemen energi. Menentukan ketika kapan melakukan *sampling* kekuatan sinyal pada WiFi *Received Signal Strength Indicator* (RSSI) dan seberapa sering harus mengumpulkan data sedemikian rupa merupakan suatu tantangan yang dihadapi dalam manajemen sumber daya. Melakukan eksplorasi terhadap data adalah konsep yang banyak digunakan dalam kebijakan pengumpulan data (A. Deligiannakis et al., 2008). Pengambilan sampel secara adaptif (*adaptive sampling*) adalah salah satu cara yang paling komprehensif dalam pengumpulan data. Dengan menyesuaikan frekuensi pengambilan sampel kekuatan sinyal pada WiFi dari waktu ke waktu, untuk



beradaptasi terhadap perubahan yang dihasilkan dari dinamika *Indoor positioning system* (IPS). Komponen lain untuk melakukan *sampling* secara adaptif adalah aktivitas pada tingkat *user* dimana pada kondisi sedang proses WiFi *scanning*. Proses tersebut menunjukkan bahwa sebelum melakukan proses *scanning*, sistem harus mengecek terlebih dahulu jika *user* bergerak secara signifikan dari posisi yang sudah diketahui sebelumnya. Contoh kasus apabila *user* dalam kondisi diam, seperti tidur atau diam dalam suatu ruangan bisa dikatakan dalam kondisi tidak bergerak untuk periode yang lebih lama dalam periode siklus pengambilan sampel, dan itu akan terjadi redundansi untuk penandaan suatu posisi.

Proses pergerakan *user* dapat dideteksi dengan mengimplementasikan *Bluetooth* (Kyohei Sugino et al., 2016), menurutnya dengan mengembangkan deteksi pergerakan pada perangkat seluler dengan menggunakan *Bluetooth* untuk memantau objek lebih baik dibandingkan sensor lainnya seperti *Infrared light* dan *ultrasonic waves*, karena sensor tersebut ada keterbatasan dengan masalah posisi dan memerlukan daya yang lebih. Menurut (Kyohei Sugino et al., 2016) deteksi gerak menggunakan *Bluetooth* ini mempunyai dua keuntungan, yang pertama detektor gerak dapat dipasang secara *flexible* dan tertutup dari orang-orang. Keuntungan kedua *Bluetooth* bekerja dengan menggunakan daya baterai kecil, karena mengirimkan data paket yang kecil termasuk hanya beberapa ID. Di samping hal tersebut sangat mudah dalam pemeliharaan *system* deteksi gerak menggunakan *Bluetooth*. Pada penelitian sebelumnya umumnya deteksi gerak menggunakan *accelerometer*.

### **3.2. Studi Literatur**

Tahap ini merupakan tahap pengumpulan informasi pengerjaan penelitian. Tahap-tahap dalam penelitian tersebut sebagai berikut :

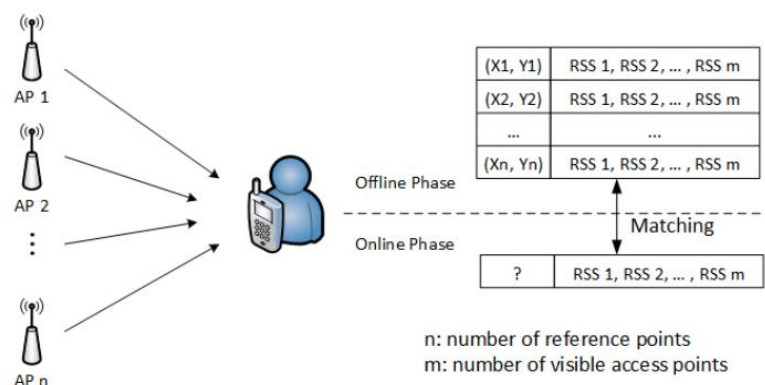
#### **3.2.1. *Fingerprint***

*Fingerprint* merupakan suatu titik penanda berfokus pada sinyal di setiap lokasi tertentu. Ini didasarkan pada asumsi bahwa untuk setiap posisi tertentu memiliki pola sinyal yang unik. Teknik berbasis *fingerprint* bekerja dengan cara

survei lokasi, dimana pola sinyal di sekitar lokasi dikumpulkan untuk membangun *database*. Karena sinyal dapat bervariasi dari waktu ke waktu ditempat tertentu, yang dicatat dalam periode waktu tertentu. Dan *point of fingerprint* merupakan suatu titik yang berisi sekumpulan penanda atau sidik jari (*fingerprint*) hasil dari *sampling* yang telah dilakukan sebelumnya pada tiap ruangan (Jun MA et al., 2008).

Menurut (Xuan Du., 2018) *Fingerprint* adalah teknik penentuan posisi yang signifikan digunakan sebagian besar oleh sistem penentuan posisi berbasis RSS. Pada tahap penentuan posisi, RSS yang secara *real-time* dari sinyal WiFi di sekitarnya diterima oleh *user* dengan perangkat kompatibel dengan WiFi dibandingkan dengan *fingerprint* yang sudah disimpan sebelumnya di *database* untuk menemukan *fingerprint* yang paling cocok, yang dilakukan oleh algoritma pemosisian. Informasi lokasi yang sesuai dari *fingerprint* paling cocok akan dikembalikan ke *user* sebagai hasil dari penentuan posisi.

Pada teknik *fingerprint* seperti pada Gambar 3.1 umumnya terbagi menjadi dua tahap: 1. tahap *offline* dan 2. tahap *online*. Pada tahap *offline* disebut juga sebagai tahap *training* dan tahap *online* disebut juga sebagai tahap penentuan posisi. Pada tahap *offline radio map* dibuat dengan menyimpan informasi tentang semua AP (Access Point) yang terlihat dan nilai RSSI dari semua titik lokasi, yang bisa dijadikan titik rujukan. Setelah mengumpulkan data untuk masing-masing lokasi kemudian dibuat *fingerprint*.

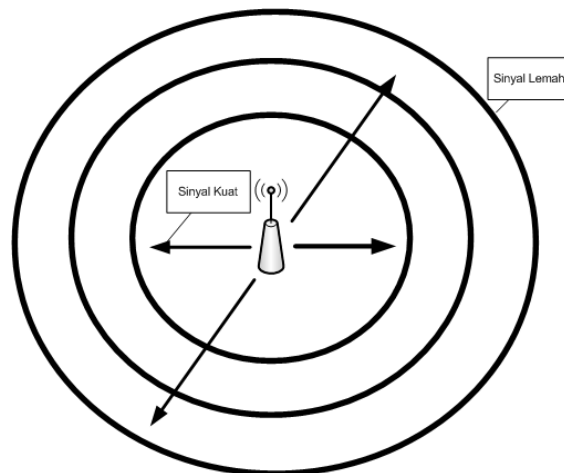


Gambar 3.1. (Xuan Du., 2018) Penentuan Posisi *Fingerprint*

### **3.2.2. Received Signal Strength Indicator (RSSI)**

RSSI digunakan untuk menilai jarak antara *mobile node* atau unit berdasarkan redaman sinyal. Keuntungan menggunakan sistem ini adalah penerapannya yang sederhana dan biaya rendah karena penerima sistem nirkabel terintegrasi secara umum dengan kecukupan pengukuran RSS, yang sudah digunakan untuk tujuan seperti kontrol energi transmisi dan juga kontrol otomatis. RSS adalah pengukuran dari suatu energi yang tersedia dalam sinyal radio yang diterima. Dalam teknik Pengukuran Berbasis Radio yang ada, RSS telah digunakan secara luas dalam *Indoor positioning system (IPS)*. RSS menentukan sifat dan karakteristik dari penanda sidik jari (*fingerprint*) dari suatu lokasi. Tegangan sirkuit diukur antara perangkat pemancar dan penerima yang bervariasi karena antarmuka jalur dan diukur dengan *Received Signal Strength Indicator (RSSI)* yang diterima (HongyuShi, 2012), (C.L.Wang et al., 2006). Ukuran jarak diperkirakan dengan menerima simpul dari simpul sumber dengan menentukan energi dari kekuatan sinyal yang diterima. Ia menggunakan faktor *path-loss* untuk mengubah RSS menjadi ukuran jarak dan bekerja berdasarkan prinsip bahwa daya sinyal berkurang jika jarak antar node meningkat. Menurut sistem standar IEEE 802.11 RSS adalah kekuatan sinyal yang diterima yang sesuai dalam lingkungan nirkabel.

Kondisi dalam ruangan menjadi sulit untuk mendeteksi *Line-of-Sight (LOS)* di antara unit-unit, RSSI (*Received Signal Strength Indicator*) sangat dipengaruhi oleh bayangan, dinding bangunan, jenis antena dan bahkan *multipath fading*, sehingga sulit untuk mengusulkan suatu model matematika *channel* yang cocok dengan propagasi nyata, dengan demikian menghasilkan estimasi yang tidak tepat. Ketika biaya rendah menjadi prioritas, maka sistem berbasis RSSI ini adalah merupakan suatu solusi terbaik untuk digunakan dalam pemosisian suatu ruangan (Amol Potgantwar et al., 2015). Ilustrasi dari RSS bisa dilihat pada Gambar 3.2 di bawah ini.



Gambar 3.2. Kekuatan Sinyal

### 3.2.3. Adaptive Sampling

*Adaptive sampling* pada dasarnya berarti bahwa secara dinamis mengubah interval *sampling* dari suatu sensor didasarkan pada kondisi pemicu tertentu. Hal itu dapat diklasifikasikan menjadi dua tipe:

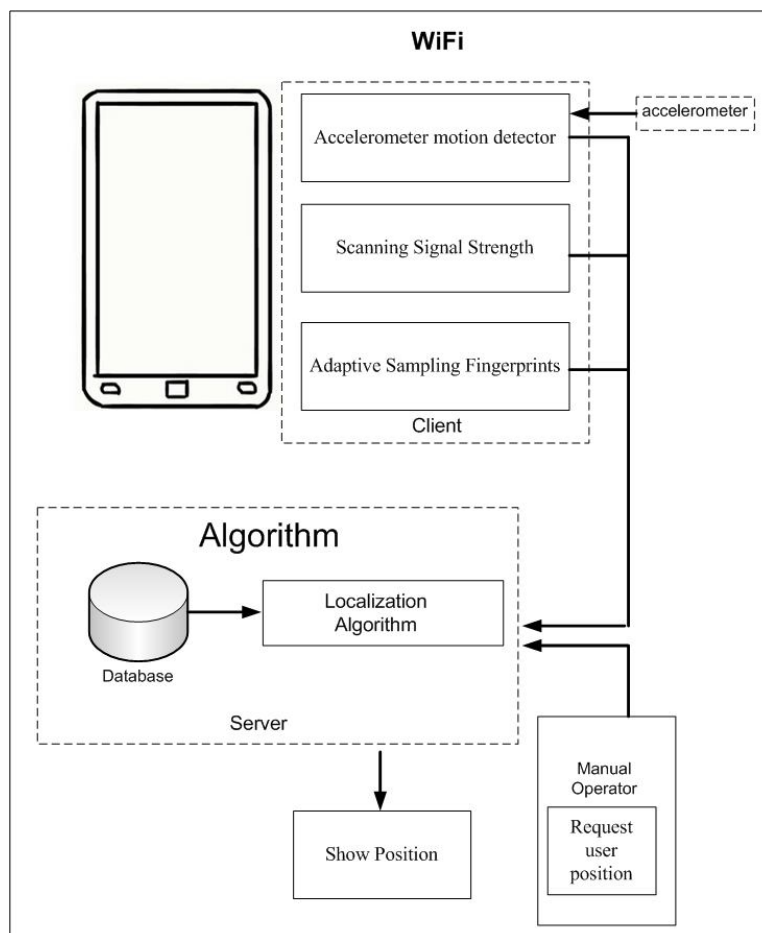
1. Pengambilan sampel dipicu oleh sensor lain, di mana seberapa sering data diambil dari satu sensor, atau bahkan diambil sama sekali, tergantung pada *output* dari sensor lain (misalnya. menunggu *input* dari *accelerometer* tentang pergerakan sebelum memulai GPS atau WiFi).
2. Pengambilan sampel dipicu oleh *output* sensor sendiri, di mana periode interval pengambilan sampel disesuaikan secara dinamis berdasarkan data yang diambil sebelumnya dari sensor yang sama. Frekuensi pengambilan sampel bertambah atau berkurang berdasarkan seberapa berguna data yang direkam sebelumnya (ditentukan oleh klasifikasi), dan kecepatan modifikasi interval biasanya didasarkan pada fungsi tertentu (linier, eksponensial, dll). Sebagai contoh, frekuensi pengambilan sampel WiFi akan berkurang ketika klasifikasi akan mendeteksi sedikit atau tidak ada perubahan dalam titik posisi yang terdeteksi dalam sampel yang direkam sebelumnya, dan meningkat ketika titik posisi yang terdeteksi bervariasi dengan masing-masing *scan* berikutnya.

Manfaat pengambilan sampel secara adaptif adalah untuk menghemat baterai yang terdiri dari kenyataan bahwa energi hanya dapat digunakan ketika data yang berguna tersedia. Sebagai contoh data yang berguna dan tidak berguna, misalnya contoh kasus untuk memutuskan seberapa sering mengambil sampel dari sensor GPS, salah satu sensor paling mahal pada perangkat bergerak. Hal yang tidak mungkin dilakukan adalah untuk terus mengambil sampel GPS dengan kecepatan yang sama, jika misalnya dua atau tiga estimasi berturut-turut akan menunjukkan bahwa *user* berada di lokasi yang sama.

Demikian pula, ketika mengambil sampel menggunakan kekuatan sinyal WiFi, tidak ada titik posisi yang terdeteksi atau tidak ada perubahan yang terdeteksi di titik posisi selama beberapa sampel, akan berarti bahwa tingkat pengambilan sampel harus turun, sementara lebih banyak titik posisi yang terdeteksi akan berarti peningkatan tingkat pengambilan sampel. Dengan cara ini, energi dapat dipakai untuk data yang benar-benar penting, sehingga meningkatkan ukuran data yang berguna untuk penelitian. Sebagai bukti efek pengambilan sampel secara adaptif adalah terhadap masa pakai baterai (Radu Călin Gătej, 2013).

### **3.3. Desain Sistem**

Pada desain sistem ini akan menjelaskan alur secara umum, bagaimana proses mulai deteksi gerak menggunakan *accelerometer* bekerja, kemudian *scanning* kekuatan sinyal, dilanjutkan dengan proses *sampling fingerprint*, kemudian diproses melalui algoritma klasifikasi, disini penulis menggunakan pendekatan *K-Nearest Neighbors* dan terakhir adalah proses menampilkan hasil dari posisi. Untuk lebih jelas gambaran umum desain sistem bisa dilihat pada Gambar 3.3 berikut di bawah.

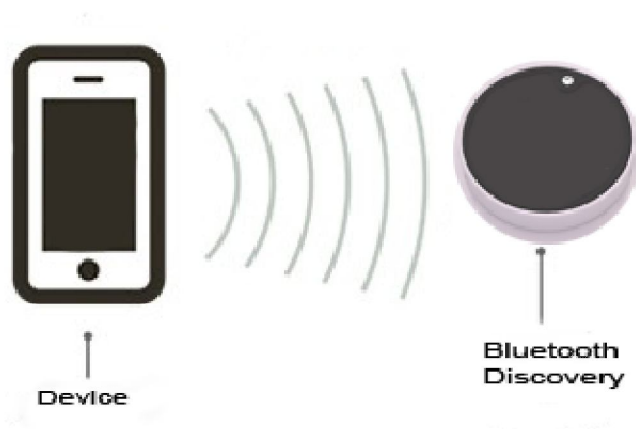


Gambar 3.3. Desain Sistem

Berdasarkan desain sistem pada Gambar 3.3, Informasi posisi *user* dapat diketahui dengan mendeteksi kekuatan sinyal WiFi yang ditangkap oleh perangkat seluler *user* pada saat itu, yang sebelumnya aplikasi pada *smartphone* mendeteksi gerakan menggunakan fungsi perangkat *accelerometer*. Begitu juga *smartphone* yang menggunakan *Bluetooth* sebagai pendeteksi gerak, *Bluetooth transmitter* mentransmisikan sinyal, dan penerima *Bluetooth* akan mengukur RSSI dari sinyal yang sesuai, ilustrasinya bisa dilihat pada gambar 3.4 yang menggambarkan *Bluetooth* sedang mentransmisikan sinyal. RSSI akan konstan apabila tidak ada penghalang diantara kedua perangkat. Jika objek berada diantara perangkat, RSSI akan berkurang dan apabila terjadi peningkatan dispersi RSSI maka hal itu dianggap ada suatu gerakan. Karena itu, ketika RSSI yang diukur jatuh di bawah nilai ambang batas (*threshold*) yang ditetapkan  $If AVG(t,T) < \theta$  (Kyohei Suginoet

al., 2016), maka proses *scanning* kekuatan sinyal WiFi akan diproses, dan selanjutnya akan dilakukan proses *sampling finger print* yang kemudian dikirim ke server untuk diproses selanjutnya. *Sampling* dari kekuatan sinyal WiFi kemudian dilakukan proses perhitungan dengan menggunakan algoritma penentuan posisi, untuk proses ini peneliti menggunakan pendekatan *K-Nearest Neighbors* untuk proses klasifikasi dan *Cluster Filtered KNN* untuk meningkatkan presisi hasilnya.

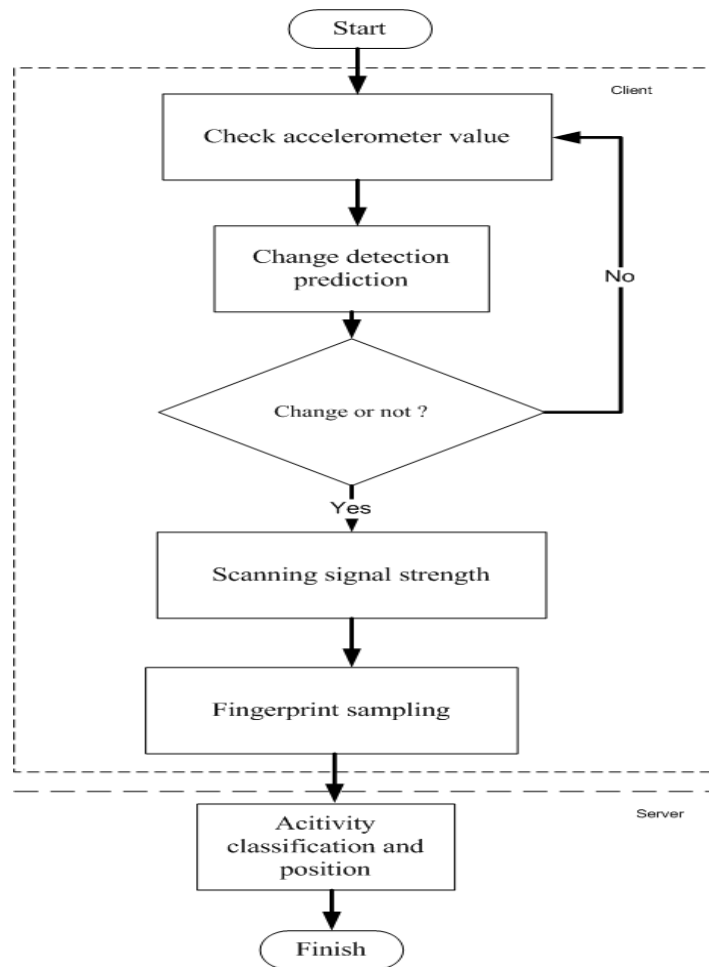
Setelah proses perhitungan *sampling*, sistem akan menampilkan hasil akhir berupa koordinat dari posisi *user*. Selanjutnya akan ditampilkan informasi posisi *user* dengan berupa *map* sebagai penerjemah secara visual dari koordinat yang informasinya bisa dipahami oleh *user*. Untuk menambah asas manfaat dari sistem, manual operator bisa melakukan *request* untuk posisi dari *user* saat ini dan kemudian sebagai *feedback* sistem akan mengirim informasi posisi *user* berupa *map* secara visual.



Gambar 3.4. Transmisi Sinyal *Bluetooth*

### 3.4. Desain Algoritma dan Implementasi

Pada tahap ini akan dijelaskan bagian alur sistem dari *Indoor positioning system (IPS)* seperti pada Gambar 3.3. Konsep umum dari IPS adalah menentukan koordinat global sebuah lokasi seperti garis bujur dan garis lintang.



Gambar 3.5. Alur Kerja Sistem

Berdasarkan Gambar 3.5, perancangan alur sistem ini adalah sebagai berikut:

1. Dalam proses *scanning* sinyal WiFi berlaku secara terus menerus, sehingga ketika proses *sampling* pada posisi yang sama akan terjadi redundansi data. Untuk mengatasi hal tersebut dibutuhkan suatu *trigger* untuk mengatur proses *scanning* yaitu *change detection*. Sistem akan mengecek kondisi perubahan tingkat fluktuasi nilai pada *accelerometer* pertama kali, seperti yang sudah dijelaskan di bab 2 sub bab 2.5, kenapa menggunakannya, karena *accelerometer* secara *default* sudah tertanam pada *smartphone* dan berbiaya rendah (Yapeng Wang et al., 2013), kemudian *change detection* akan mendeteksi apakah ada suatu perubahan nilai. Dan mengecek apakah *user* telah bergerak dari posisi yang sudah ditandai atau POF.



2. Jika *change detection* mendeteksi suatu perubahan nilai pada *accelerometer*, maka sistem akan melakukan proses *scanning signal strength* dari WiFi, dan apabila tidak mendeteksi suatu perubahan tidak dilakukan proses *scanning*.
3. Proses selanjutnya adalah melakukan *sampling* secara adaptif terhadap kekuatan sinyal WiFi.
4. Setelah proses *sampling* dilakukan pada *client*, sistem pada server akan melakukan proses klasifikasi dan melakukan update posisi berupa informasi koordinat yang akan ditampilkan berupa *map* posisi kepada *user*.

#### **3.4.1. Deteksi Pergerakan User Menggunakan Bluetooth**

Deteksi pergerakan adalah sebuah teknik yang digunakan untuk mendeteksi dan mengetahui apakah suatu *user* sedang dalam kondisi bergerak atau dalam kondisi diam. Air, kaca dan metode lainnya dapat menyerap sinyal *Bluetooth*, meskipun sinyal ini dapat menembus plastik atau bahan lainnya. Sistem biasanya menghindari dari air dan kaca, karena mereka dapat menghambat koneksi nirkabel. Tubuh manusia sekitar 60% adalah air dan mereka dapat menghalangi sinyal *Bluetooth*. Ketika seseorang berdiri diantara pemancar dan penerima *Bluetooth*, maka nilai dari RSSI akan berkurang. Akibatnya kita dapat menggunakan *Bluetooth transmitter* untuk menentukan redaman RSSI dan mencapai lokalisasi tanpa membawa perangkat.

Jika perangkat *Bluetooth* diatur dalam jarak 1 m, perangkat penerima akan mengukur sekitar -50 dB RSSI dari sinyal *Bluetooth*. Ketika seseorang berdiri di antara perangkat, maka sinyal tidak sepenuhnya terlindungi, karena sinyal *Bluetooth* adalah *omni directional*. Tetapi nilai RSSI akan berkurang dan dispersi distribusi RSSI akan meningkat. Akibatnya, detektor gerakan dapat membedakan bahwa seseorang berada diantara perangkat *Bluetooth*.

Dispersi RSSI akan meningkat ketika seseorang bergerak. Oleh karena itu, ada kemungkinan kesalahan karena variasi RSSI, dan hal ini dapat menyebabkan detektor gerakan untuk menentukan bahwa seseorang bergerak, yang ketika itu pada kenyataannya tidak terjadi. Jadi peneliti menggunakan nilai rata-rata dari beberapa sampel yang diambil pada menit terakhir. Sistem mengukur RSSI pada

interval satu detik: itu adalah RSSI ( $t$ ) pada waktu  $t$ . Peneliti mendefinisikan  $AVG(t,T)$  sebagai sampel rata-rata untuk  $T$  detik dari waktu  $t$ . Perubahan RSSI dapat diamati ketika seseorang bergerak di antara perangkat *Bluetooth*. Kemudian membandingkan nilai RSSI yang diperoleh ketika seseorang diam dan ketika seseorang tidak memiliki nilai ambang batas  $\theta$ . Jika  $AVG(t,T) < \theta$ , maka sistem akan menentukan bahwa seseorang tersebut adalah diam (Kyohei Suginoet al., 2016). Dan berikut spesifikasi detail dari *Bluetooth* yang akan dipergunakan.

**Tabel 3.1** Tabel Spesifikasi *Bluetooth*

Spesifikasi Bluetooth	USB Dongle
Versi	V2.0 & 1.2 Compliant
Supporting Profile	Jaringan, Dial-Up, Fax, LAN Access & Headset
Sistem Operasi	Windows 2000, Xp, windows 7/8/10
Interface	USB Compliance
Symbol Rate	3 Mbps
Jangkauan	0 - 100 m

### 3.4.2. Deklarasi Perijinan Akses Komponen

Dalam perijinan akses komponen perlu di definisikan terlebih dahulu pada *AndroidManifest.xml*, sesuai desain sistem yang dibutuhkan untuk jalannya sebuah aplikasi. Dalam hal ini hak akses yang dibutuhkan adalah akses penggunaan WiFi dan *Bluetooth*. Gambar 3.5 berikut menjelaskan mengenai hak akses yang dibutuhkan dalam pengembangan sistem.

```
<uses-permission android:name=" android.permission.BLUETOOTH_ADMIN" />
<uses-permission android:name=" android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
```

Gambar 3.6. Deklarasi Perijinan Akses Komponen

#### 3.4.2.1. Proses WiFi dan *Scanning*

Bagian ini menjelaskan mengenai implementasi bagaimana aplikasi memperoleh data berupa kekuatan sinyal dari setiap WiFi. Mulai dari inisialisasi

Wifi sebagai bagian dari *service* yang sebelumnya sudah dijelaskan dari hak akses komponen. Selanjutnya tahap *scanning* kekuatan sinyal dari WiFi, untuk mendaftarkan informasi WiFi yang ada di sekitar, Penjelasan dari *event* tersebut menghasilkan Pseudocode 3.1 berikut ini.

1	String connectivity_context = Context.WIFI_SERVICE;
2	WifiManager wifi = (WifiManager) getSystemService(connectivity_context);
3	scanObject = new StartScan(wifi);
4	<b>public class</b> StartScan {
5	WifiManager m_wifi;
6	List<ScanResult> m_result;
7	<b>public</b> StartScan(WifiManager wifi) {
8	m_wifi = wifi;
9	<b>if</b> (!m_wifi.isWifiEnabled()) {
10	<b>if</b> (m_wifi.getWifiState() != WifiManager.WIFI_STATE_ENABLING) {
11	m_wifi.setWifiEnabled( <b>true</b> );
12	}
13	}
14	}
15	<b>public</b> List<ScanResult> scan() {
16	m_wifi.startScan();
17	m_result = m_wifi.getScanResults();
18	<b>return</b> m_result;
19	}
20	<b>public int</b> getOnRSSI(String bssid) {
21	List<ScanResult> result = scan();
22	Iterator<ScanResult> iscan = result.iterator();
23	<b>while</b> (iscan.hasNext()) {
24	ScanResult nextResult = iscan.next();
25	<b>if</b> (bssid.compareTo(nextResult.BSSID) == 0)
26	<b>return</b> nextResult.level;
27	}
28	
29	}
30	}

Pseudocode 3.1. Inisialisasi WiFi dan *Scanning*

### 3.4.2.2. Proses Sampling Fingerprint

*Sampling* dilakukan dengan mengubah data yang ada di komponen *List data stream* ke dalam *database*. Lebih jelasnya berikut Pseudocode *sampling fingerprint*.

1	<b>private</b> XmlSerializer m_serializer = Xml.newSerializer(); <b>private</b> Map<String, String>m_bssid_ssid <b>private</b> Map<String, List<String>>m_bssid_rssi; String tag = "store";
2	<b>public</b> StoreInfo(int fren, long time, Calendar start, String position, String scenery) <b>throws</b> IllegalArgumentException, IllegalStateException {
3	m_allResult = <b>new</b> ArrayList<List<ScanResult>>();
4	m_bssid_ssid = <b>new</b> HashMap<String, String>();
5	m_bssid_rssi = <b>new</b> HashMap<String, List<String>>();
6	}
7	<b>public void</b> writeXml() <b>throws</b> IOException { File v_xmlfile = <b>new</b> File(Environment.getExternalStorageDirectory() .getPath() + "SignalStrength/" + m_position + "_" + m_startTime.getTime().toString() + ".xml");
8	v_xmlfile.createNewFile();
9	OutputStream v_os = <b>new</b> FileOutputStream(v_xmlfile, <b>true</b> );
10	m_serializer.text(WriteFile.transTimeToString(m_startTime));
11	getApAllRSSI();
12	Iterator<String> iBssid = m_bssid_ssid.keySet().iterator();
13	<b>while</b> (iBssid.hasNext()) {
14	String bssid = iBssid.next();
15	m_serializer.startTag( <b>null</b> , "rssi");
16	List<String> thisApRssi = m_bssid_rssi.get(bssid);
17	Iterator<String> iRssi = thisApRssi.iterator();
18	<b>while</b> (iRssi.hasNext()) {
19	m_serializer.text(iRssi.next() + "\r\n");
20	}
21	m_serializer.endTag( <b>null</b> , "rssi");
22	m_serializer.endTag( <b>null</b> , "ap");
23	}
24	m_serializer.endTag( <b>null</b> , "allapiinfo");
25	m_serializer.endTag( <b>null</b> , "scan");
26	m_serializer.text("\r\n");
27	m_serializer.endDocument();
28	v_os.flush();
29	v_os.close();

30	}

### Pseudocode 3.2. *Sampling Fingerprint*

#### 3.4.2.3. Proses Deteksi *Bluetooth*

Proses koneksi *Bluetooth* menggunakan *BleManager* yang berfungsi untuk menangani konektivitas dan menginisialisasi perangkat setelah terkoneksi. Berikut Pseudocode proses deteksi *Bluetooth*.

1	<b>protected</b> E mCallbacks; <b>private</b> Handler mHandler; <b>private</b> List<BluetoothGatt> mBluetoothGatt; <b>private</b> Context mContext; <b>private boolean</b> mUserDisconnected;
2	<b>public</b> List<BluetoothGatt> getBluetoothGatt() { return mBluetoothGatt; }
3	<b>private</b> BroadcastReceiver mBondingBroadcastReceiver = new BroadcastReceiver() {
4	@Override
5	<b>public void</b> onReceive( <b>final</b> Context context, <b>final</b> Intent intent) {
6	<b>if</b> (Build.VERSION.SDK_INT >= Build.VERSION_CODES.JELLY_BEAN_MR2) {
7	<b>final</b> BluetoothDevice device =
8	intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
9	<b>final int</b> bondState =
10	intent.getIntExtra(BluetoothDevice.EXTRA_BOND_STATE, -1);
11	<b>final int</b> previousBondState =
12	intent.getIntExtra(BluetoothDevice.EXTRA_PREVIOUS_BOND_STATE, -1);
13	// Skip other devices <b>if</b> (mBluetoothGatt == null    mBluetoothGatt.size() == 0) {
14	return;
15	}
16	<b>boolean</b> haveInclude = <b>false</b> ; <b>int</b> index = -1;
17	<b>for</b> ( <b>int</b> i = 0; i < mBluetoothGatt.size(); i++) {
18	<b>if</b> (device.getAddress().equals(mBluetoothGatt.get(i).getDevice(). getAddress())) {
19	index = i;
20	haveInclude = <b>true</b> ;
21	}
22	}
23	<b>if</b> (!haveInclude) {
24	return;
25	}
	<b>switch</b> (bondState) {
	<b>case</b> BluetoothDevice.BOND_BONDING:

26	<code>mCall backs. onBondingRequired();</code>
27	<code>break;</code>
28	<code>case BluetoothDevice. BOND_BONDED:</code>
29	<code>mCall backs. onBonded();</code>
30	<code>// Start initializing again.</code> <code>if(index!=-1) {</code>
31	<code>mBluetoothGatt. get(index). discoverServices();</code>
32	<code>}</code>
33	<code>break;</code>
35	<code>};</code>
36	<code>public BLEManager(final Context context) {</code> <code>    mContext = context;</code> <code>    mHandler = new Handler();</code> <code>    mUserDisconnected = false;</code> <code>    mBluetoothGatt = new ArrayList&lt;&gt;();</code> <code>    // Register bonding broadcast receiver</code> <code>    context. registerReceiver(mBondingBroadcastReceiver, new</code> <code>    IntentFilter(BluetoothDevice. ACTION_BOND_STATE_CHANGED));</code>
37	<code>context. registerReceiver(mPairingRequestBroadcastReceiver, new</code> <code>    IntentFilter("android.bluetooth.device.action.PAIRING_REQUEST"</code> <code>    "/*BluetoothDevice. ACTION_PAIRING_REQUEST*/));</code> <code>}</code>

Pseudocode 3.3. Proses Deteksi *Bluetooth*

### 3.4.3. Prediksi *Change Detection*

Prediksi *change detection* merupakan suatu proses dalam memprediksi adanya suatu perubahan nilai tertentu pada sekumpulan data. Tujuan dari Prediksi *change detection* adalah untuk memprediksi apakah adanya suatu perubahan dari satu proses kejadian ke proses kejadian yang lain dan seberapa besar perubahannya tersebut, hal ini dapat dijadikan sebagai indikator untuk *trigger* dari perubahan itu.

Proses yang mendasari sifat prediksi *change detection* adalah untuk penghematan energi daya baterai dengan memprediksi apakah ada atau tidaknya suatu perubahan data yang akan dijadikan sebagai *trigger* untuk melakukan proses *scanning* kekuatan sinyal WiFi, penjelasannya dapat dilihat pada Gambar 3.7.

Pada penelitian ini prediksi *change detection* menggunakan sejumlah kumpulan data dari perubahan posisi sebagai studi kasus untuk *scanning* dari kekuatan sinyal WiFi. Proses prediksi *change detection* akan dimulai dengan melakukan *sampling* data *Bluetooth*. Interval waktu tertentu pada *sampling* data

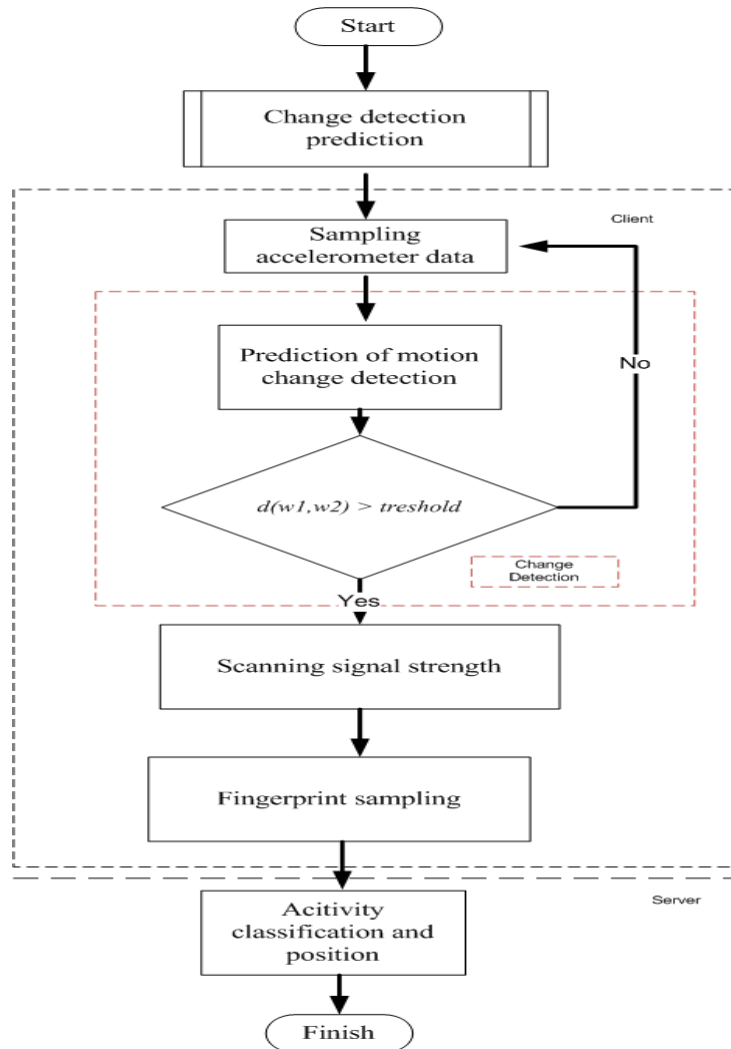
menunjukkan *sampling event mobile user* dalam interval waktu tertentu. Ketika sistem mendapatkan  $S$  sampel data, perubahan yang terjadi dapat diprediksi menggunakan pendekatan kedekatan rasio  $S/2$  prior data dan  $S/2$  current data atau pendekatan *threshold* dengan *change detection* formula pada Paper (Raquel Sebastião et al., 2007). Penjelasan prediksi *change detection* dapat dilihat pada Pseudocode 3.4. sesuai referensi (Daniel Kifer et al., 2004).

1	for $i = 1 \dots k$ do
2	$c_0 \leftarrow \emptyset$
3	Window $_{1,i} \leftarrow$ first $m_{1,i}$ points from time $c_0$
4	Window $_{2,i} \leftarrow$ next $m_{2,i}$ points in stream
5	end for
6	While not at end of stream do
7	for $i = 1 \dots k$ do
8	Slide Window $_{2,i}$ by 1 point
9	if $d(\text{Window}_{1,i}, \text{Window}_{2,i}) > \alpha_i$ then
10	$c_0 \leftarrow$ current time
11	Report change at time $c_0$
12	Clear all windows and GOTO step 1
13	end if
14	end for
15	end while

Pseudocode 3.4. *Change Detection*

Keterangan pada Pseudocode 3.4 diatas menunjukkan bahwa proses menjalankan pengulangan sebanyak  $k$  populasi berjalan secara paralel untuk setiap parameter  $(m_{1,i}, m_{2,i}, \alpha_i)$ . Sistem juga membutuhkan suatu fungsi  $d$ , yang berfungsi untuk mengukur perbedaan antara dua sampel, dan tiga parameter  $\{(m_{1,1}, m_{2,1}, \alpha_1), \dots, (m_{1,k}, m_{2,k}, \alpha_k)\}$ . Angka dari  $m_{1,i}$  dan  $m_{2,i}$  menentukan ukuran pasangan kedua *windows*  $(X_i, Y_i)$ . *Windows*  $X_i$  merupakan *baseline* dan berisi nilai  $m_{1,i}$  menunjukkan aliran yang terjadi setelah perubahan terakhir yang terdeteksi. Setiap *windows*  $Y_i$  merupakan *sliding window* yang berisi  $m_{2,i}$  terbaru dari item-item dalam aliran data. Setelah perubahan terdeteksi, *windows*  $Y_i$  berisi titik  $m_{2,i}$  dari aliran yang mengikuti *windows*  $X_i$ . Selanjutnya menggeser *windows*  $Y_i$  satu langkah ke depan setiap kali item baru muncul dalam aliran. Pada setiap pembaruan tersebut, kita harus memeriksa apakah  $d(X_i, Y_i) > \alpha_i$ . Apabila terjadi jarak  $> \alpha_i$ , maka dianggap adanya suatu perubahan dan kemudian mengulangi

seluruh prosedur dengan  $X_i$  yang berisi nilai  $m_{1,i}$  pertama, menunjukkan setelah adanya perubahan (Daniel Kiferet al., 2004).



Gambar 3.7. Alur kerja proses prediksi *motion change detection*

### 3.5. Pengujian

Pengujian dilakukan secara langsung menggunakan tiga perangkat WiFi pada ruangan, untuk menguji proses *scanning* sinyal WiFi, proses pengolahan data kekuatan sinyal WiFi yang dikirim ke server dan informasi posisi setelah server melakukan pengolahan data yang dikirim dari *client*, serta menguji perbandingan bukti efek pengambilan sampel secara adaptif dengan menggunakan *Bluetooth transmitter* sebagai pemancar terhadap penghematan masa pakai baterai. Dan akan diuji juga seberapa nilai yang menentukan tetap dan berubahnya



terhadap nilai RSSI dari *Bluetooth* ketika suatu objek terdeteksi sedang bergerak atau diam dan seberapa besar perubahan datanya. Berikut contoh Gambar penempatan perangkat WiFi pada suatu ruangan, seperti pada Gambar 3.8.



Gambar 3.8. Penempatan Perangkat WiFi

### 3.5.1. Prediksi *Motion Change Detection*

Proses pertama kali yang dilakukan dari prediksi *motion change detection* adalah dimulai dengan melakukan *sampling* data terhadap *signal strength* RSSI dari *Bluetooth* dalam interval waktu tertentu. Dalam penelitian ini direncanakan beberapa skenario untuk membedakan antara objek yang sedang bergerak dan objek yang dalam keadaan diam dengan memasang *Bluetooth transmitter* atau sejenisnya berupa *dongle* yang tertancap pada *personal computer*. Yang pertama kali dilakukan adalah dengan melakukan percobaan pendahuluan untuk mengukur redaman RSSI oleh objek atau manusia. Secara khusus peneliti akan meneliti perubahan RSSI yang disebabkan oleh pergerakan tubuh manusia. Eksperimen akan dilakukan di ruangan kelas kosong. Peneliti akan menggunakan *Bluetooth transmitter* sebagai pemancar yang tertancap pada PC. Untuk mengidentifikasi status subjek tersebut apakah bergerak atau tidak untuk kalibrasi RSSI, dan menandainya dengan notasi *m1* sebagai kelompok data *prior sampling* dan *m2*

sebagai kelompok data *current sampling* kemudian menandai  $\alpha$  sebagai *threshold*. Berdasarkan Pseudocode 3.4. referensi (Daniel Kiferet al., 2004) di atas. Sekumpulan data  $m1$  akan di simpan pada *window*  $X_i$  dan  $m2$  akan disimpan pada *window*  $y_i$ . Selanjutnya dibutuhkan pengukur perbedaan antara  $X_i$  dan  $Y_i$  dengan menggunakan fungsi  $d$ . Kemudian dicek apakah  $d(X_i, Y_i) > \alpha$ , apabila terlihat suatu perubahan nilai yang melebihi *threshold* maka dianggap sebagai adanya suatu perubahan atau gerak. Berikut beberapa skenario pengujian *change detection* dengan perkiraan waktu pengujian selama 120 menit dan tiap 10 menit dilakukan pengecekan perubahan level baterai.

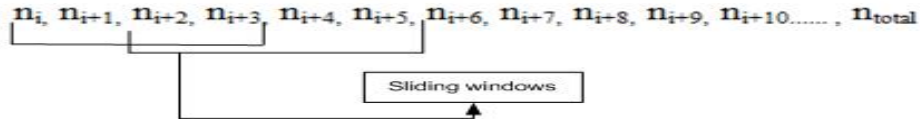
**Tabel 3.2** Skenario Pengujian

No	Skenario Pengujian
1	Pengujian dalam kondisi diam 100%
2	Pengujian 10% dalam kondisi gerak dari total waktu pengujian
3	Pengujian 30% dalam kondisi gerak dari total waktu pengujian
4	Pengujian 50% dalam kondisi gerak dari total waktu pengujian
5	Pengujian 75% dalam kondisi gerak dari total waktu pengujian
6	Pengujian 100% dalam kondisi gerak
7	Berapa % penghematan terhadap efisiensi energi
8	Akurasi lokasi pada saat reporting

### 3.5.2. Variasi S Sliding Window terhadap Akurasi Prediksi *Motion Change Detection*

Deretan data *Bluetooth* yang diperoleh dalam bentuk RSSI mempresentasikan beberapa deretan data kekuatan sinyal pada *Bluetooth* yang dilakukan dalam proses pertama kali dalam interval waktu tertentu. Proses prediksi *motion change detection* dilakukan tidak hanya terjadi sekali selama interval waktu tersebut, tetapi dilakukan pengulangan beberapa kali. Setiap proses prediksi, data kekuatan sinyal *Bluetooth* diambil dalam jumlah tertentu. Ketika melakukan kembali proses prediksi, maka sejumlah data RSSI dari *Bluetooth* berikutnya akan diambil dengan cara proses *sliding windows*, yaitu dengan

menggeser data awal untuk memulai proses berikutnya sejumlah data  $S$ . Informasi proses *sliding window* seperti pada gambar berikut Gambar 3.9.



Gambar 3.9. Proses *Sliding Windows*

### 3.5.3. Entropy based Change Detection

Sebagai pengukuran informasi berdasarkan paper (Raquel Sebastião et al., 2007), peneliti mengadaptasi *Entropy* sebagai pengukuran *change detection*. *Entropy* antara probabilitas perbedaan absolut mengukur dispersi dari Perbedaan Distribusi dan ditentukan oleh persamaan berikut:

$$H(p||q) = -\sum_i |q(i) - p(i)| * \log_2(|q(i) - p(i)|) \quad (3.1)$$

Dimana  $q_i$  dan  $p_i$  mewakili probabilitas dari *point bin i* dari *window* saat ini dan probabilitas dari bin referensi koresponden. Dari sifat-sifat persamaan di atas, kita dapat memperoleh batas:  $0 \leq H(p||q) \leq 2$ . Nilai  $H(p||q)$  yang lebih kecil, sesuai dengan dispersi yang lebih kecil antara distribusi dua variabel.

### 3.5.4. Kullback-Leibler based Change Detection

*Entropy* Relatif adalah salah satu yang paling umum cara mewakili jarak antara dua distribusi (Tamraparni Dasuet et al., 2006). Juga dikenal sebagai jarak atau divergensi Kullback-Leibler, dan ini mempunyai fungsi untuk mengukur jarak antara dua distribusi probabilitas dan sehingga dapat digunakan untuk menguji perubahan. Berikut ini adalah *reference window* dengan  $p_i$  probabilitas empiris, dan *sliding window* dengan probabilitas  $q_i$ , dan jarak KL berikut adalah:

$$KL(p||q) = -\sum_i p(i) * \log_2 p(i)/q(i) \quad (3.2)$$

Divergensi KL adalah non negatif dan asimetris dan semakin tinggi nilainya, semakin berbeda juga distribusi kedua variabel. Nilai jarak yang lebih tinggi mewakili distribusi yang lebih jauh.

### 3.5.5. Cosine Distance based Change Detection

Selain informasi yang membandingkan jarak antara dua distribusi probabilitas, peneliti juga membandingkan sudut di antara keduanya. Berdasarkan *paper* (Raquel Sebastião et al., 2007) Jarak cosinus diturunkan dari titik *product* dari dua vektor dan ditentukan oleh persamaan berikut:

$$C(p||q) = - \frac{\sum_i p(i)*q(i)}{||p|| ||q||} \quad (3.3)$$

Peneliti menganggap definisi ini untuk memiliki nilai non negatif dan menjamin bahwa  $0 \leq C(q||p) \leq 2$ . Ukuran ini adalah simetris dan nilai yang lebih rendah bisa diartikan bahwa dua distribusi lebih dekat.

### 3.5.6. Silent Zone

*Silent zone* merupakan rentang nilai pada alur *silent - movement mode* (Duc-tan et al., 2015), yang mana rentang nilai tersebut merupakan rentang toleransi atas *noise* pada data. Berikut contoh data dengan simulasi menggunakan *silent mode*. Setelah dilakukan pengambilan data untuk *silent zone* selama 10 kali percobaan dengan 10 raw data, maka didapatkan total raw data sebanyak 100 data, seperti pada Gambar 3.10. berikut di bawah :

	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9	i=10
Percobaan1	-80	-77	-75	-75	-71	-66	-60	-62	-70	-63
Percobaan2	-59	-60	-69	-58	-59	-58	-59	-61	-69	-71
Percobaan3	-76	-77	-71	-64	-65	-67	-72	-73	-65	-59
Percobaan4	-61	-59	-67	-76	-77	-67	-64	-65	-67	-72
Percobaan5	-73	-65	-68	-70	-73	-71	-62	-64	-60	-58
Percobaan6	-61	-59	-67	-57	-63	-54	-62	-60	-66	-76
Percobaan7	-70	-73	-66	-60	-64	-60	-62	-61	-64	-61
Percobaan8	-63	-66	-61	-66	-61	-61	-60	-63	-65	-67
Percobaan9	-64	-63	-68	-63	-68	-62	-60	-62	-64	-67
Percobaan10	-68	-67	-70	-62	-66	-67	-62	-66	-69	-62

Gambar 3.10. Raw Data Pada Bluetooth

	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9	i=10
Percobaan1	0,9546	0,5371	1,3756	1,6785	1,0999	0,6668	1,5182	0,8489	1,3378	1,0430
Percobaan2	1,0430	0,1474	1,3855	0,3844	0,9152	1,3332	0,8902	1,2398	0,9968	0,6666
Percobaan3	0,9004	1,1166	0,4866	1,0881	1,5403	1,0411	1,2791	1,0436	1,8286	0,6514
Percobaan4	2,5371	2,7061	2,1653	1,8665	1,4991	0,6323	1,4753	1,1825	2,2365	1,8718
Percobaan5	1,5645	1,8224	1,5439	1,3843	1,3741	1,0487	0,8843	1,2594	1,2435	1,5143
Percobaan6	2,7540	2,2236	1,2565	0,9125	1,3164	1,6873	1,2825	0,5409	0,7164	1,1391
Percobaan7	1,4528	1,5552	1,5565	1,4904	1,8033	1,9368	1,3587	0,5605	0,7033	1,4530
Percobaan8	1,6581	1,4399	1,3286	1,4236	1,3888	0,7194	0,6723	0,1150	0,8587	0,6693
Percobaan9	0,6068	0,8528	0,8996	0,7919	0,7715	0,6735	0,8023	1,1647	1,2361	0,7948
Percobaan10	0,8232	1,3620	1,1138	0,5899	0,7760	0,4833	0,7326	1,2281	1,0409	0,9584

Gambar 3.11. Raw Data Pada Accelerometer

Dari semua data percobaan pada Gambar 3.10 dan Gambar 3.11, akan dilakukan *training* dengan cara mengambil nilai selisih perubahan percepatannya, dari nilai selisih tersebut ditentukan nilai *absolute* dengan menggunakan rumus di bawah berikut :

$$Difference = |X_{i+1} - X_i| \quad (3.4)$$

Berdasarkan rumus (3.4) di atas didapatkan hasil selisih *absolute* seperti gambar berikut di bawah

T=9	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9
training1	3	2	0	4	5	6	2	8	7
training2	1	9	11	1	1	1	2	8	2
training3	1	6	7	1	2	5	1	8	6
training4	2	8	9	1	10	3	1	2	5
training5	8	3	2	3	2	9	2	4	2
training6	2	8	10	6	9	8	2	6	10
training7	3	7	6	4	4	2	1	3	3
training8	3	5	5	5	0	1	3	2	2
training9	1	5	5	5	6	2	2	2	3
training10	1	3	8	4	1	5	4	3	7

Gambar 3.12. Selisih *Absolute* Hasil Training *Raw Data* Pada *Bluetooth*

T=9	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8	i=9
training1	0,41753	0,83851	0,30285	0,57854	0,43310	0,85137	0,66931	0,48897	0,29482
training2	0,89559	1,23806	1,00105	0,53078	0,41800	0,44300	0,34958	0,24299	0,33017
training3	0,21619	0,63000	0,60148	0,45223	0,49927	0,23806	0,23554	0,78499	1,17717
training4	0,16897	0,54081	0,29878	0,36742	0,86671	0,84292	0,29273	1,05391	0,36467
training5	0,25791	0,27858	0,15958	0,01020	0,32537	0,16442	0,37511	0,01591	0,27078
training6	0,53038	0,96707	0,34402	0,40392	0,37090	0,40481	0,74161	0,17552	0,42266
training7	0,10241	0,00125	0,06611	0,31291	0,13357	0,57815	0,79822	0,14283	0,74972
training8	0,21820	0,11130	0,09503	0,03481	0,66937	0,04712	0,55730	0,74372	0,18938
training9	0,24608	0,04672	0,10771	0,02039	0,09795	0,12877	0,36237	0,07143	0,44134
training10	0,53887	0,24817	0,52392	0,18602	0,29263	0,24932	0,49542	0,18715	0,08254

Gambar 3.13. Selisih *Absolute* Hasil Training *Raw Data* Pada *Accelerometer*

Untuk *threshold* atau ambang batas nilai untuk *Silent Zone*, dipilih nilai selisih yang paling besar dari semua training selama beberapa kali percobaan. Dari contoh data berdasarkan Gambar 3.12, didapatkan nilai kecil adalah 5 dan nilai terbesarnya yaitu 11 dan pada Gambar 2.13, didapatkan nilai terkecil 0,37511 dan nilai terbesarnya 1,23806. Untuk mendapatkan batas bawah dan batas atas dari *silent zone* untuk *Bluetooth*, terlebih dahulu kita harus mencari standar deviasi dari persebaran datanya.

$$S^2 = \frac{\sum_{i=1}^n x_i^2 - n\bar{X}^2}{n-1} \quad (3.5)$$

Setelah dilakukan perhitungan standar deviasi ( $\sigma$ ) untuk data training yang sudah dilakukan pengambilan selisih *absolute*, maka didapatkan nilai **1,08012345**, untuk Bluetooth dan 0,29931 untuk *accelerometer* kemudian dengan rumus berikut ini :

$$\begin{aligned} \text{Batas bawah} &= \mu - \sigma \\ \text{Batas atas} &= \mu + \sigma \end{aligned} \quad (3.6)$$

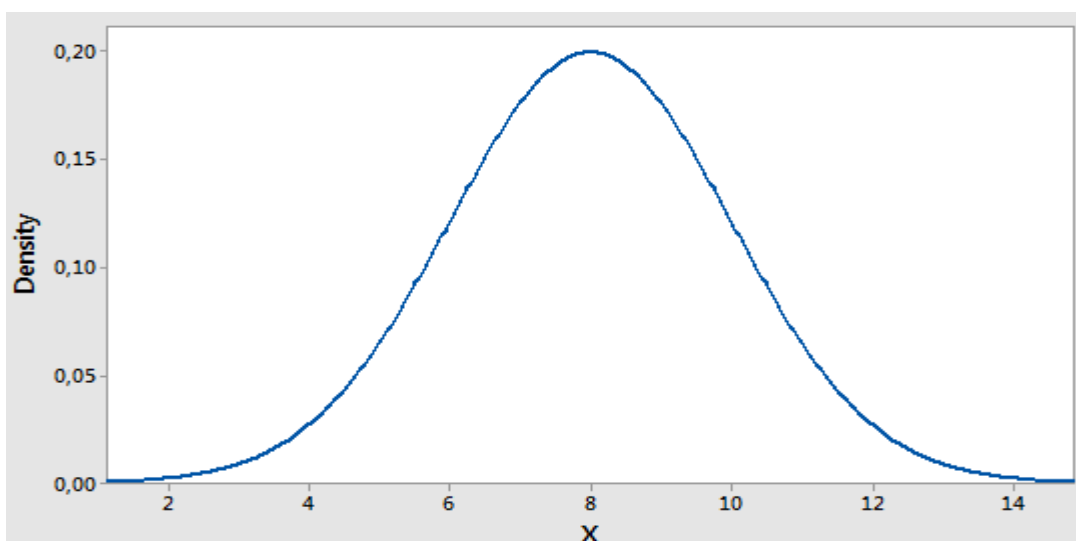
Kita akan mencari batas atas dan batas bawah dari *Silent Zone* data *Bluetooth*. Sehingga didapatkan batas bawah *Silent Zone* : **6** untuk Bluetooth, **0,51917** untuk *accelerometer* dan batas atas *Silent Zone* : **10** untuk Bluetooth dan **1,11780** untuk *accelerometer*. Setelah melakukan transformasi nilai batas bawah dan batas atas selanjutnya menggunakan rumus :

$$Z = \frac{x - \mu}{\sigma} \quad (3.7)$$

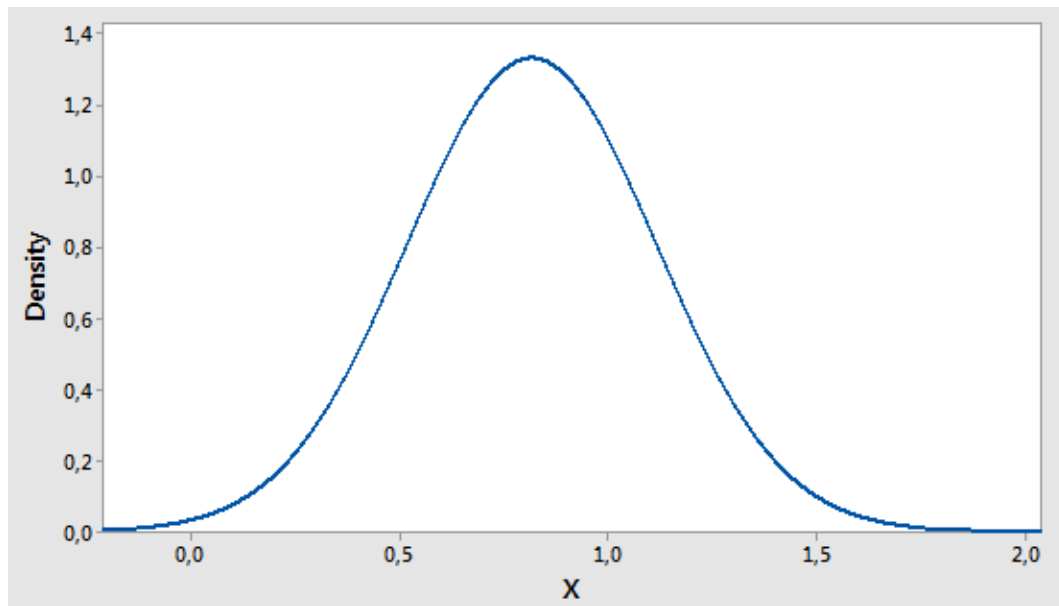
Sehingga didapatkan  $Z_{\min}$  dan  $Z_{\max}$  yaitu +1 dan -1, maka kita bisa mencari grafik persebaran probabilitas kumulatifnya dengan memanfaatkan persamaan Distribusi Gaussian :

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.8)$$

Dari rumus persamaan di atas dapat diilustrasikan dengan Gambar 3.14 dan Gambar 3.15 seperti berikut:



Gambar 3.14. Grafik Persebaran Probabilitas Kumulatif *Raw Data Bluetooth*



Gambar 3.15. Grafik Persebaran Probabilitas Kumulatif *Raw Data Accelerometer*

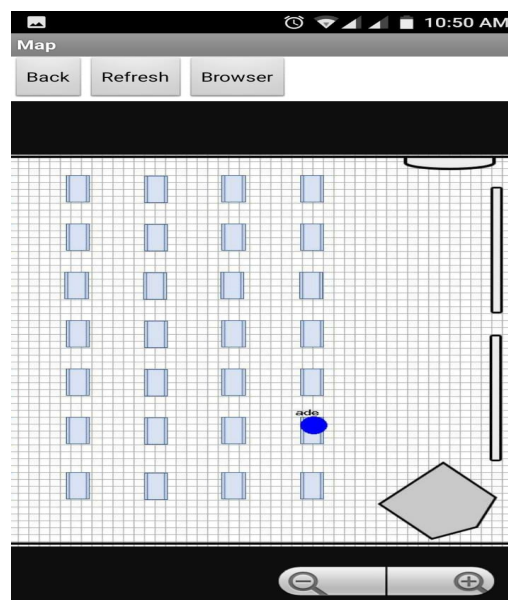
**Tabel 3.3** Ringkasan analisis probabilitas kumulatif

	<i>Bluetooth</i>	<i>Accelerometer</i>
<b>Data min</b>	5	0,37511
<b>Data max</b>	11	1,23806
<b>Mean</b>	8	0,81848
<b>Stdev</b>	2	0,29931
<b>Batas bawah</b>	6	0,51917
<b>Batas atas</b>	10	1,11780

### 3.5.6. *Scanning* Sinyal WiFi

Proses *scanning* merupakan fungsi untuk memperoleh data berupa kekuatan sinyal di beberapa posisi yang telah ditentukan. Pada tahap pertama *training*, semua referensi titik di *map* dengan koordinat fisik yang diketahui (x,y) ditandai dengan ID lokasi. Semua titik pada Gambar 3.13 dipilih sebagai titik

survei untuk eksperimen. Jarak yang lebih kecil antara titik *anchor* dapat meningkatkan akurasi sistem, tetapi itu tidak berarti bahwa ketepatannya juga akan meningkat, karena titik *anchor* yang berbeda mungkin memiliki WiFi *fingerprint* yang sama (Wasiq Waqar et al., 2016). Dan juga ketika kita memilih titik *anchor* lebih dekat satu sama lain akan membuat fase *training* lebih sulit. Tidak ada pedoman standar untuk ukuran grid sesuai pada Gambar 3.9 yang seharusnya. Sebelum melakukan proses *scanning*, *user* harus melakukan *login* terlebih dahulu tujuannya untuk menentukan hasil *scanning user* tersebut. Untuk *Login layout* bisa dilihat pada Gambar 3.17.

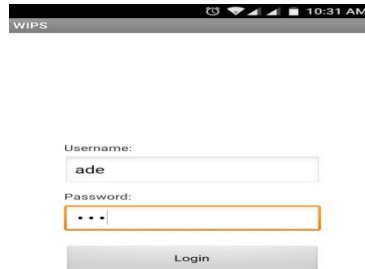


Gambar 3.16. *Map* Titik Lokasi

Setelah *user* berhasil *login*, sistem akan melakukan Proses *scanning* dan mengumpulkan data, kemudian menandai AP WiFi dan menyimpannya ke *database*. Ketika AP *transmitter* diproses oleh NIC (*network interface card*), banyak informasi mengenai AP tertentu yang bisa ditampilkan. Setiap AP memiliki alamat MAC yang unik. Bersama dengan alamat MAC, informasi RSSI juga disimpan. Tidak ada pedoman umum tentang seberapa banyak jumlah *scanning* yang diperlukan untuk mengumpulkan data, tetapi dalam penelitian ini pada area eksperimental sebanyak 3 AP yang akan dilakukan untuk uji coba



scanning. Gambar 3.18 menunjukkan *raw* data yang dikumpulkan untuk satu titik survei tersebut.



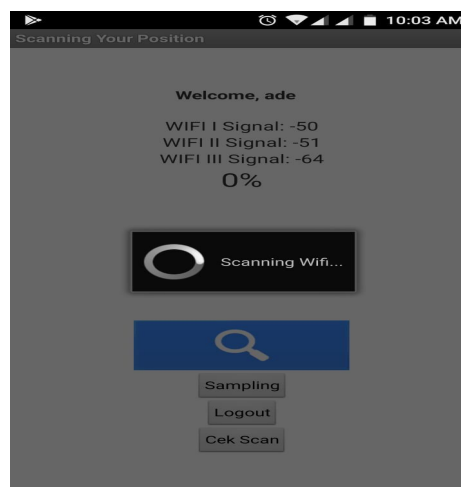
Gambar 3.17. *Login User*

Berikut di bawah merupakan Gambar 3.18 yang menunjukkan informasi data mentah pada suatu titik survei.



Gambar 3.18. *Raw Data WiFi Scanning*

Berikut dibawah merupakan Gambar tampilan *Scanning WiFi*, seperti pada Gambar 3.19.



Gambar 3.19. *Scanning WiFi*

### **3.5.7. WiFi Fingerprint**

Setelah mengumpulkan informasi *raw* data dari WiFi, pada tahap kedua adalah melakukan WiFi *fingerprint*. *Fingerprint* RSSI hasil ekstraksi *raw* data dari setiap titik survei atau referensi akan ditampilkan ke dalam suatu statistik. Pada penelitian akan menguji fluktuasi dan varian RSSI dari suatu AP tertentu yang terlebih dahulu kita definisikan. Tetapi dalam penelitian akan fokus pada penghematan daya baterai. Seperti yang dijelaskan sebelumnya MAC merupakan pengidentifikasian unik untuk setiap *wireless network interface card* (NIC). Peneliti menggunakannya untuk membedakan antara WiFi AP yang berbeda dalam jangkauan, selama pengumpulan beberapa *scanning* dilakukan di lokasi yang sama. Setiap WiFi *scanning* berisi nilai RSSI instan dari masing-masing AP. Karena nilai RSSI berfluktuasi, perlu untuk mengambil nilai rata-rata. Jumlah kemunculan AP selama periode *sampling* juga merupakan bagian dari *fingerprint*.

### **3.5.8. Estimasi Posisi**

Pada tahap penentuan posisi, pengukuran WiFi langsung dilakukan pada lokasi penelitian dan sistem kemudian akan mengecek ke *database* untuk melakukan *fingerprint matching*. Satu kali WiFi *scanning* selama tahap penentuan posisi dapat menghasilkan *matching* yang kurang baik, karena mungkin kekurangan data RSSI yang tidak cukup. Menurut (Yan Luo et al., 2011) kesalahan penentuan posisi akan lebih besar jika jumlah *scanning* kurang dari 4 kali, tetapi dengan 4 kali *scanning* atau lebih, akurasi dari pemosisian akan stabil. Dalam penelitian berencana akan melakukan proses *scanning* sebanyak 5 kali dengan tujuan untuk mendapatkan hasil penentuan posisi yang maksimal. Langkah selanjutnya adalah menghitung perkiraan posisi yang paling mungkin dengan mencocokkan hasil *scanning* dengan semua *fingerprint* yang sudah tersimpan dalam *database*.

### **3.5.9. Efisiensi Energi Baterai**

Untuk mengetahui pengaruh dari proses *scanning* sistem terhadap energi baterai, peneliti akan melakukan uji coba dengan menjalankan aplikasi selama waktu yang telah ditentukan, kemudian mengamati penurunan energi baterai.

Dan akan menguji juga seberapa besar energi yang terpakai ketika perangkat seluler menjalankan *Bluetooth* dan WiFi. Kinerja baterai akan dievaluasi menggunakan konsumsi baterai rata-rata per jam, diukur pada skala 0 hingga 100, 100 adalah mewakili pengisian baterai maksimum. Rencana pengujian ini akan dilakukan selama 120 menit dengan melakukan pengamatan setiap 10 menit, dengan tujuan untuk mengetahui seberapa besar jumlah penurunan energi dari baterai.

#### **3.5.10. Presentasi Posisi**

*User* akan tertarik dengan informasi posisi akhir yang diperkirakan oleh sistem. Dalam penelitian ini berencana mencoba beberapa kemungkinan untuk menguji pada situasi dimana beberapa titik *anchor* memiliki probabilitas yang sangat mendekati untuk posisi *user*. Untuk menghasilkan posisi yang paling mungkin sebagai perkiraan posisi, peneliti akan menguji dengan menjalankan sistem yang akan dibangun di perangkat *user*. Dengan demikian hasil dari pengujian tersebut berupa koordinat (x,y) yang akan ditampilkan dengan berupa *map* yang didesain sesuai dengan kondisi tempat penelitian.

#### **3.5.11. Evaluasi**

Dalam penelitian ini, peneliti berencana akan menjelaskan metodologi, uji coba, skenario dan hasil dari eksperimental. Tujuan dari eksperimental utama adalah untuk mengukur penghematan energi dari suatu proses *scanning* untuk *sampling* terhadap kekuatan sinyal WiFi secara adaptif. Serta untuk mengetahui seberapa besar manfaat informasi gerak terhadap perubahan RSSI dari suatu *Bluetooth* untuk penentuan posisi *user* di lingkungan dalam ruangan dan pengaruhnya terhadap penggunaan energi baterai.

### 3.6. Jadwal Penelitian

**Tabel 3.4** Jadwal Kegiatan Penelitian

No.	Kegiatan	Bulan												
		I			II			III			IV			
1.	Studi Literatur	■	■	■										
2.	Desain Sistem		■	■	■									
3.	Implementasi Sistem					■	■	■	■					
4.	Uji Coba dan Analisa Hasil								■	■	■			
5.	Penyusunan Laporan								■	■	■	■	■	■

*[Halaman ini sengaja dikosongkan]*

## BAB 4

### HASIL DAN PEMBAHASAN

Pada bab 4 ini dijelaskan tahapan penelitian mengenai pengembangan mekanisme *change detection* untuk efisiensi energi, tahapan pengujian yang dilakukan dan analisis hasil uji coba yang akan dipresentasikan.

#### 4.1. Tahapan Penelitian

Penelitian ini dilakukan pada lingkungan uji coba yang real. Secara umum penelitian pengembangan mekanisme *change detection* untuk efisiensi energi dilakukan melalui beberapa tahapan berikut :

1. Tahap pertama, pada tahapan ini merupakan tahap perancangan dan implementasi sistem dimulai dengan sampling data masukan dari kekuatan sinyal RSSI dari *Bluetooth*. Kekuatan sinyal *Bluetooth* akan dilakukan training dengan cara mengambil nilai selisih perubahan percepatannya pada fluktuasi pada masing-masing nilai RSSI, dari nilai selisih tersebut ditentukan nilai *absolute* nya. Kemudian dari deretan nilai *absolute* tersebut diambil nilai maksimal yang akan dijadikan sebagai nilai *threshold* atau ambang batas nilai untuk *Silent Zone*, dipilih nilai selisih yang paling besar dari semua training selama beberapa kali percobaan. Hal tersebut adalah untuk mengatur proses *scanning* yaitu *change detection*. Fungsi *change detection* adalah untuk mendeteksi suatu perubahan nilai RSSI, kemudian sistem akan melakukan proses *scanning* kekuatan sinyal dari WiFi, dan apabila tidak terdeteksi suatu perubahan maka tidak ada proses *scanning* WiFi.
2. Tahap kedua, pada tahapan ini akan dilakukan skenario uji coba yang meliputi skenario pengujian fungsionalitas dan performa. Untuk skenario uji coba fungsionalitas berfokus pada *change detection* untuk mengetahui aktivitas pada perubahan pada proses *scanning* WiFi. Selanjutnya untuk skenario uji coba performa akan melihat pada penurunan daya baterai, seberapa besar proses tersebut menggunakan daya baterai dalam interval waktu yang sudah ditentukan.

3. Tahap ketiga, pada tahapan ini akan dilakukan skenario uji coba secara menyeluruh pada fungsionalitas dan performa pada lingkungan *real device* sesuai dengan skenario uji coba yang telah dirancang sebelumnya.

#### **4.2. Langkah-langkah Uji Coba**

Pengujian pada penelitian ini meliputi parameter penggunaan daya baterai yang merupakan pengujian yang akan dijadikan tolok ukur seberapa besar sistem yang berjalan menggunakan energi. Tujuan penelitian ini adalah penghematan energi terhadap proses *scanning* kekuatan sinyal WiFi. Selanjutnya menentukan skenario pengujian dan melakukan analisis terhadap hasil uji coba untuk mengukur sejauh mana tingkat keberhasilan dari sistem.

##### **4.2.1. Skenario Pengujian Performa**

Skenario pengujian performa ini dilakukan untuk melihat seberapa jauh keberhasilan yang dicapai terhadap tujuan penelitian. Beberapa skenario pengujian performa sebagai berikut.

###### **1. Uji Coba Penurunan daya baterai**

Pada pengujian ini dilakukan untuk mengetahui seberapa besar pengaruh dari pengembangan mekanisme *change detection* terhadap penurunan daya baterai dan penghematan daya baterai yang dicapai dengan adanya mekanisme *change detection*. Pengujian dilakukan dengan membandingkan penurunan daya baterai dari proses *scanning* normal, artinya sebelum proses *scanning* ini dikembangkan dengan adanya penambahan mekanisme *change detection* dan proses *scanning* dengan adanya pengembangan mekanisme *change detection*. Pengujian keduanya dilakukan masing-masing selama 120 menit. Tiap perubahan pada interval 10 menit akan dicatat, sebagai informasi *level* baterai yang sudah dipergunakan dalam proses. Pada uji coba penurunan baterai akan dilakukan beberapa skenario seperti berikut :

- a. Pengujian IPS *scanning* periodik dengan delay 3 detik selama 120 menit.
- b. Pengujian dalam kondisi diam 100%
- c. Pengujian 10% dalam kondisi gerak dari total waktu pengujian
- d. Pengujian 30% dalam kondisi gerak dari total waktu pengujian

- e. Pengujian 50% dalam kondisi gerak dari total waktu pengujian
- f. Pengujian 75% dalam kondisi gerak dari total waktu pengujian
- g. Pengujian 100% dalam kondisi gerak
- h. Pengujian Berapa persen penghematan untuk efisiensi energi.

## 2. Uji Coba Akurasi Change Detection

Pada pengujian menjelaskan mengenai akurasi change detection yang dihasilkan oleh sistem. Pengujian ini akan diukur keakuratan secara riil dan secara sistem dengan cara menguji kondisi bergerak secara riil.

## 3. Uji Coba Akurasi Lokasi

Pada pengujian menjelaskan mengenai akurasi lokasi yang dihasilkan oleh sistem. Pengujian ini akan diukur hasil dari reporting sistem dengan lokasi fisik untuk mengetahui ukuran sebenarnya.

### 4.2.2. Skenario Pengujian Fungsionalitas

Skenario pengujian fungsionalitas ini dilakukan untuk melihat tingkat kesesuaian dengan aturan yang telah dibuat. Beberapa skenario pengujian fungsionalitas sebagai berikut.

#### 1. Uji Coba *Scanning* Sinyal *Bluetooth*

Pengujian proses *scanning* sinyal *Bluetooth* bertujuan untuk memperoleh data berupa kekuatan sinyal *Bluetooth*, yang pada proses selanjutnya akan diolah sebagai nilai *threshold* pada *silent zone*.

#### 2. Uji Coba Respon Sensor *Accelerometer*

Pengujian respon sensor *accelerometer* bertujuan untuk memperoleh data berupa fluktuasi data yang dihasilkan oleh sensor, yang pada proses selanjutnya akan diolah sebagai nilai *threshold* pada *silent zone* sama halnya dengan *Bluetooth*.

#### 3. Uji Coba *Scanning* Sinyal WiFi

Pengujian proses *scanning* sinyal WiFi bertujuan untuk memperoleh data berupa kekuatan sinyal di beberapa posisi yang telah ditentukan. Nilai kekuatan sinyal RSSI dari WiFi ini merupakan data sebagai penentuan lokasi pada proses



selanjutnya. Pada tahap pertama *training*, semua referensi titik di *map* dengan koordinat fisik yang diketahui (x,y) ditandai dengan ID lokasi.

#### 4. Uji Coba Hasil WiFi *Scanning*

Pengujian pada WiFi *scanning* merupakan ekstraksi *raw* data RSSI dari setiap titik survei lokasi. Pada uji coba ini akan menguji fluktuasi dan varian RSSI dari suatu AP sesuai kode MAC tertentu yang terlebih dahulu kita definisikan, untuk membedakan antara WiFi AP yang berbeda dalam jangkauan, selama pengumpulan beberapa *scanning* dilakukan di lokasi yang sama.

### 4.3. Hasil Uji Coba dan Analisis

Pada penelitian ini, uji coba dilakukan dengan data masukan dari kekuatan sinyal *Bluetooth*, *Accelerometer* sebagai *raw* data untuk proses *change detection* dan kekuatan sinyal WiFi untuk menentukan posisi lokasi. Pengambilan data IPS periodik dengan delay 3 detik dan berulang dari *device*. *Change detection* juga dilakukan berulang dengan tujuan untuk mengetahui sistem dalam keadaan diam atau sedang bergerak. Apabila diketahui *user* dalam kondisi bergerak, maka *trigger* untuk menjalankan proses *scanning* WiFi akan bekerja dan apabila diketahui *user* dalam kondisi diam, sistem tidak melakukan aktivitas *scanning*. Proses tersebut bisa dilihat hasilnya sesuai pada sub bab berikut.

#### 4.3.1. Hasil Uji Coba Performa

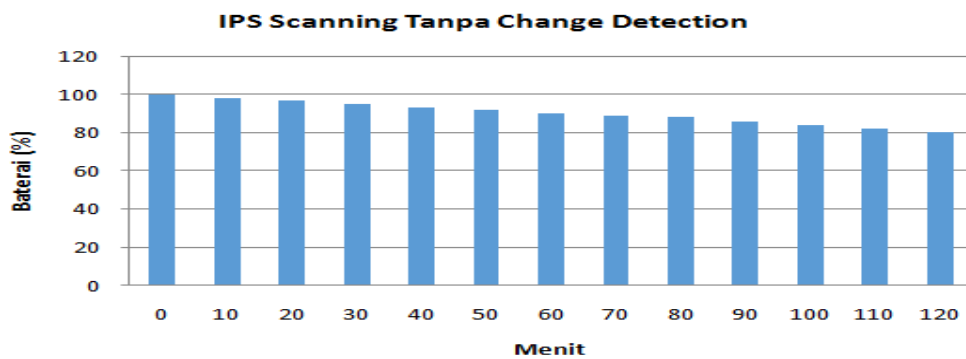
Hasil uji coba performa merupakan keluaran dari uji coba sistem yang dipresentasikan dalam bentuk gambar, grafik maupun tabel untuk mempermudah pemahaman mengenai hasil uji coba performa. Uji coba performa yang dilakukan dalam penelitian ini meliputi pengujian penurunan baterai seperti berikut pengujian dalam kondisi diam 100%, pengujian 10% dalam kondisi gerak dari total waktu pengujian, pengujian 30% dalam kondisi gerak dari total waktu pengujian, pengujian 50% dalam kondisi gerak dari total waktu pengujian, pengujian 75% dalam kondisi gerak dari total waktu pengujian, pengujian 100% dalam kondisi gerak, berapa % penghematan terhadap efisiensi energi dan akurasi lokasi pada saat reporting. Uji coba penurunan daya baterai bertujuan untuk

mengukur besarnya daya baterai dari kombinasi kondisi diam dan gerak dan berjalannya proses *change detection*.

### 1. Hasil Uji Coba Penurunan daya baterai

Berikut hasil pengujian penurunan daya baterai sesuai dengan skenario yang sudah dijelaskan di atas

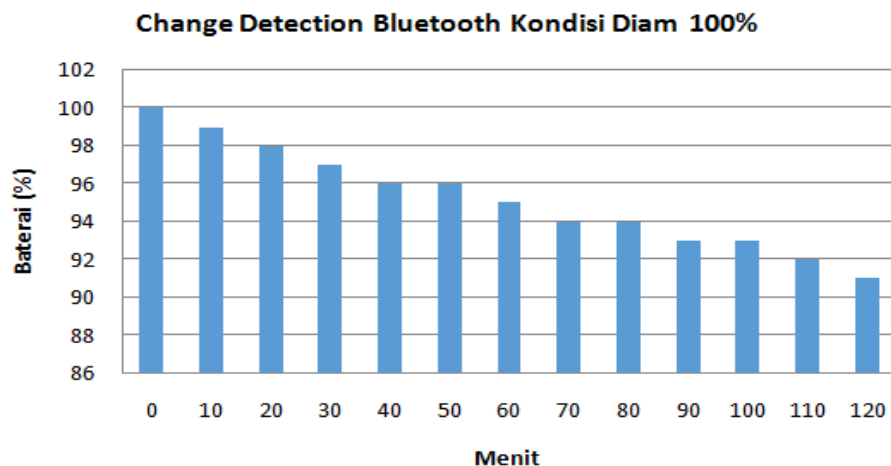
- a. Hasil pengujian IPS *scanning* periodik dengan delay 3 detik



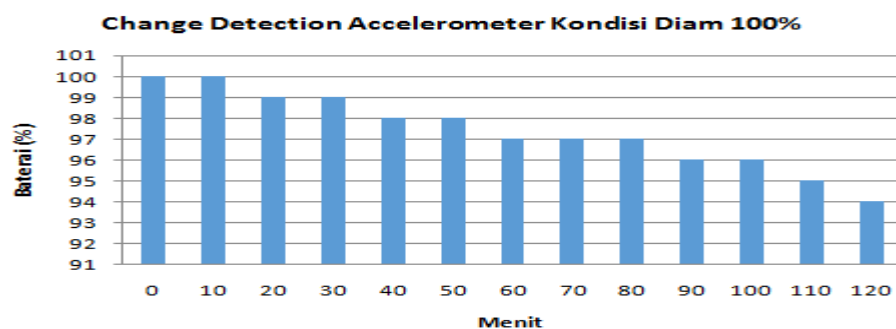
Gambar 4.1. Penurunan daya baterai IPS *scanning* tanpa *change detection*

Analisa pada Gambar 4.1 menjelaskan bahwa proses pengujian *scanning* tanpa adanya proses *change detection* untuk pengujian selama 120 menit, menunjukkan pada level baterai 80%. Pengecekan dilakukan tiap interval 10 menit. Rata-rata penurunan baterai selama 120 menit untuk IPS *scanning* tanpa *change detection* adalah 1,63% dan untuk level baterai cenderung terasa penurunannya.

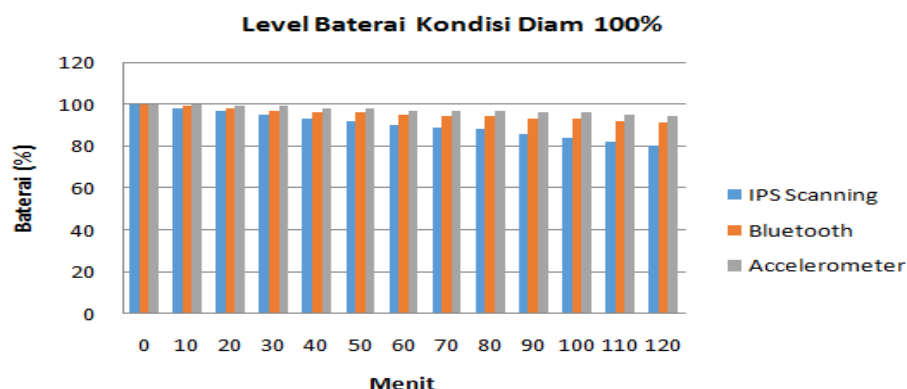
- b. Hasil pengujian dalam kondisi diam 100%



Gambar 4.2. Penurunan daya baterai kondisi diam 100% dengan *Bluetooth*



Gambar 4.3. Penurunan daya baterai kondisi diam 100% dengan *Accelerometer*

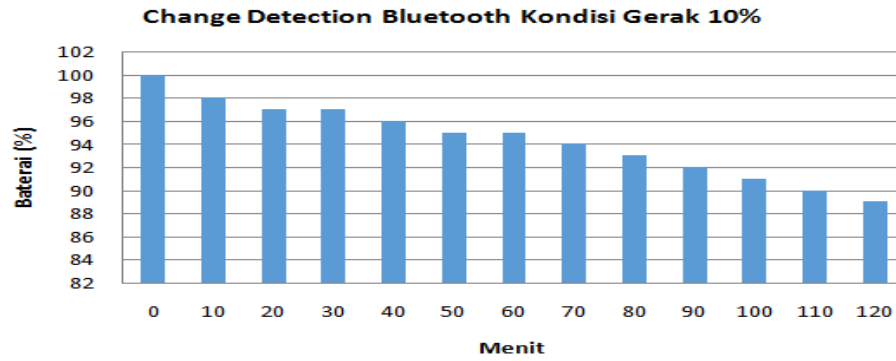


Gambar 4.4. Perbandingan penurunan daya baterai kondisi diam 100% dengan IPS periodik

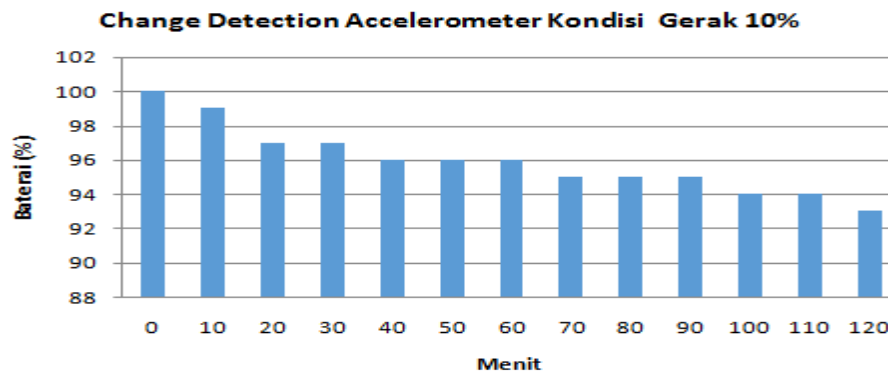
Analisa pengujian dalam kondisi diam 100%, pada Gambar 4.2 penurunan level baterai selama 120 menit menunjukkan pada level 91%. Penurunannya terasa adanya penghematan daya. Rata-rata penurunan daya baterai pada Gambar 4.2 selama 120 menit adalah 0,82%. Pengujian pada menit ke 40, 50, 70, 80, 90, 100 posisi baterai pada level yang sama. Analisa pada Gambar 4.3 penurunan level baterai selama 120 menit menunjukkan pada level 94%, penurunannya stabil dan tidak tampak adanya penurunan karena terasa adanya penghematan daya. Rata-rata penurunan daya pada Gambar 4.3 selama 120 menit adalah 0,5%. Pengujian pada menit ke 20, 30, 40, 50, 60, 70, 80, 90, 100 posisi baterai pada level yang sama. Gambar 4.4 menunjukkan perbandingan pengujian dalam kondisi diam 100% antara IPS scanning tanpa *change detection*, IPS scanning dengan *change detection* menggunakan *Bluetooth* dan *accelerometer* didapatkan penghematan energi baterai rata-rata sebesar

4,923% dengan *change detection* menggunakan *Bluetooth* dan 7,076% dengan *change detection* menggunakan *accelerometer*.

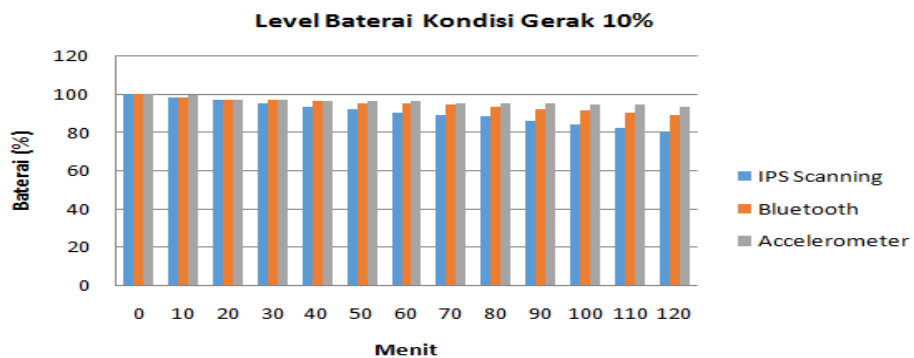
c. Hasil pengujian 10% dalam kondisi gerak



Gambar 4.5. Penurunan daya baterai kondisi gerak 10% dengan *Bluetooth*



Gambar 4.6. Penurunan daya baterai kondisi gerak 10% dengan *Accelerometer*

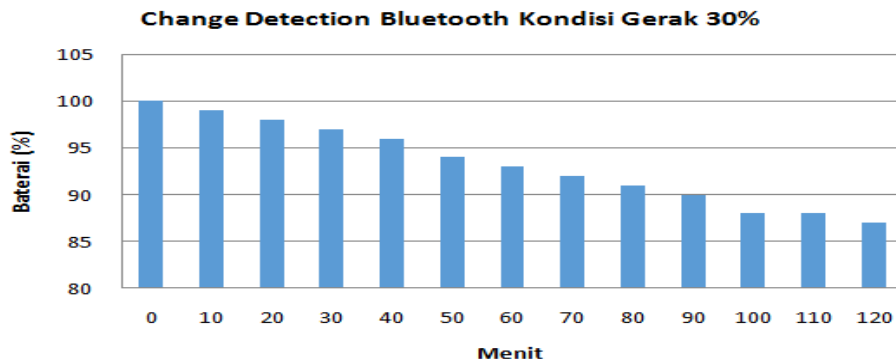


Gambar 4.7. Perbandingan penurunan daya baterai kondisi gerak 10% dengan IPS periodik

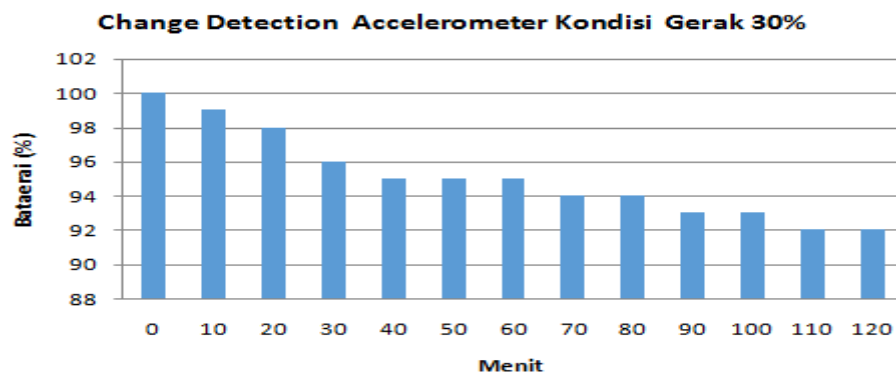
Analisa pengujian dalam kondisi gerak 10%, pada Gambar 4.5 penurunan level baterai selama 120 menit menunjukkan pada level 89%. Rata-rata

penurunan daya baterai pada Gambar 4.2 selama 120 menit adalah 0,91%. Pengujian pada menit ke 20, 30, 50, 60 posisi baterai pada level yang sama. Analisa pada Gambar 4.6 penurunan level baterai selama 120 menit menunjukkan pada level 93%, penurunannya stabil dan tidak tampak adanya penurunan karena terasa adanya penghematan daya. Rata-rata penurunan daya pada Gambar 4.6 selama 120 menit adalah 0,5% sama seperti pada pengujian dalam kondisi diam 100%. Pengujian pada menit ke 20, 30, 50, 60, 70, 80, 90, 100, 110 posisi baterai pada level yang sama. Gambar 4.7 menunjukkan perbandingan pengujian dalam kondisi gerak 10% antara IPS scanning tanpa *change detection*, didapatkan penghematan energi baterai sebesar 4,076% dengan *change detection* menggunakan *Bluetooth* dan 5,615% dengan menggunakan *accelerometer*.

d. Hasil pengujian 30% dalam kondisi gerak

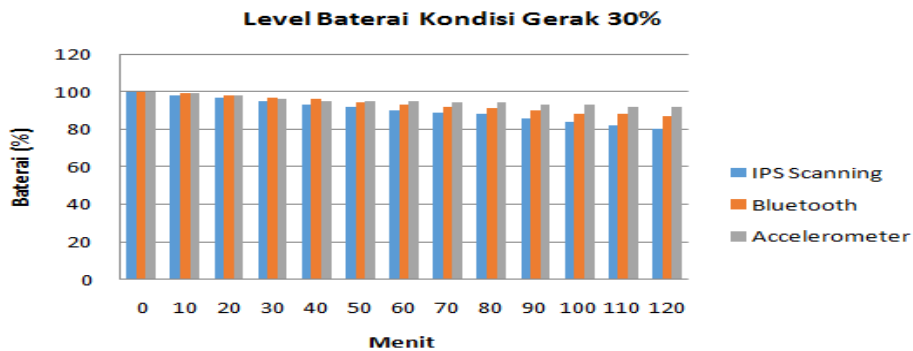


Gambar 4.8. Penurunan daya baterai kondisi gerak 30% dengan *Bluetooth*



Gambar 4.9. Penurunan daya baterai kondisi gerak 30%

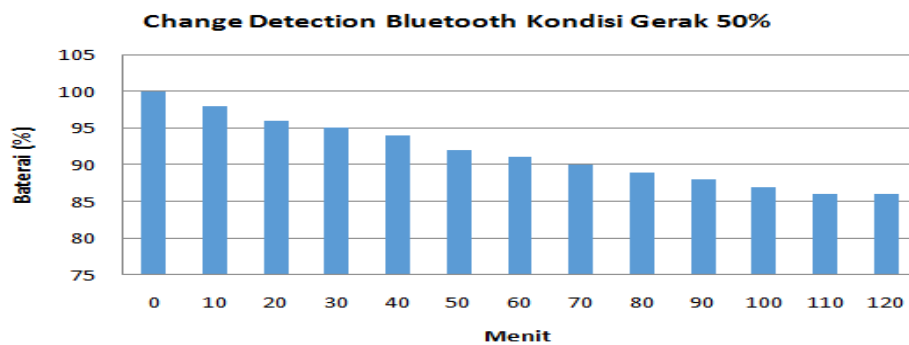
dengan *Accelerometer*



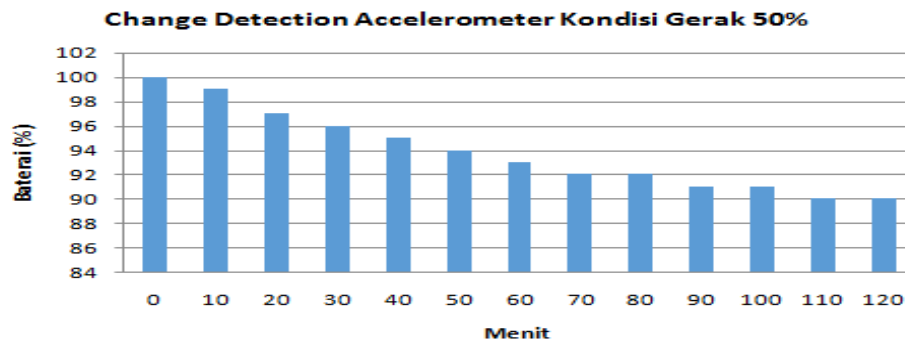
Gambar 4.10. Perbandingan penurunan daya baterai kondisi gerak 30% dengan IPS periodik

Analisa pengujian dalam kondisi gerak 30%, pada Gambar 4.8 penurunan level baterai selama 120 menit menunjukkan pada level 87%. Rata-rata penurunan daya baterai pada Gambar 4.8 selama 120 menit adalah 1%. Pengujian pada menit ke 100, 110 posisi baterai pada level yang sama. Analisa pada Gambar 4.9 penurunan level baterai selama 120 menit menunjukkan pada level 92%, penurunannya stabil rata-rata penurunan daya pada Gambar 4.9 selama 120 menit adalah 0,7%. Pengujian pada menit ke 40, 50, 60, 70, 80, 90, 100, 110, 120 posisi baterai pada level yang sama. Gambar 4.10 menunjukkan perbandingan pengujian dalam kondisi gerak 30% antara IPS scanning tanpa *change detection*, didapatkan penghematan energi baterai sebesar 3% dengan *change detection* menggunakan *Bluetooth* dan 4,769% dengan menggunakan *accelerometer*.

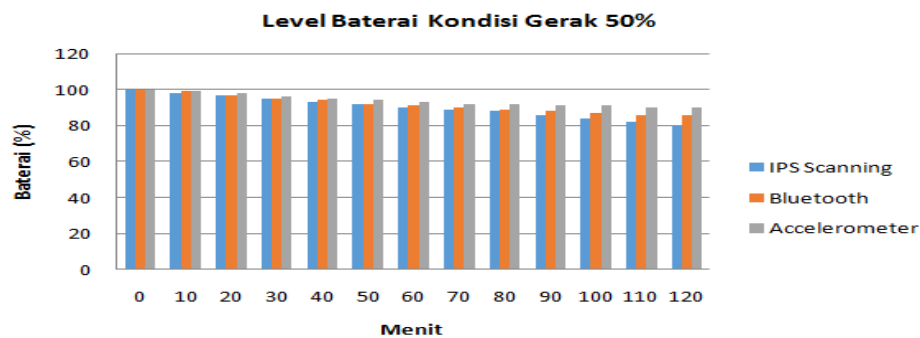
e. Hasil pengujian 50% dalam kondisi gerak



Gambar 4.11. Penurunan daya baterai kondisi gerak 50% dengan *Bluetooth*



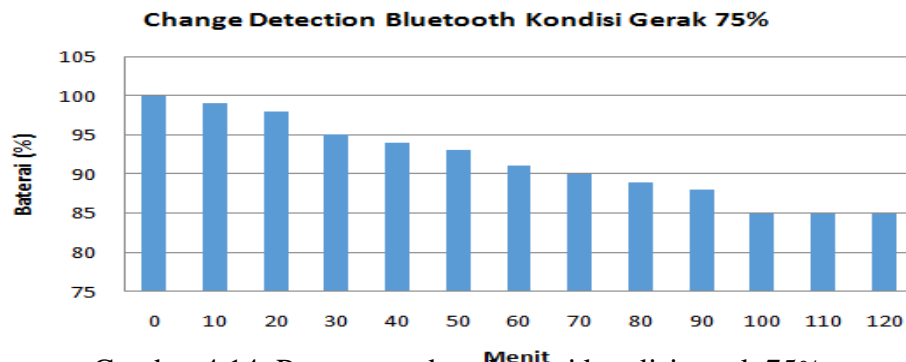
Gambar 4.12. Penurunan daya baterai kondisi gerak 50% dengan *Accelerometer*



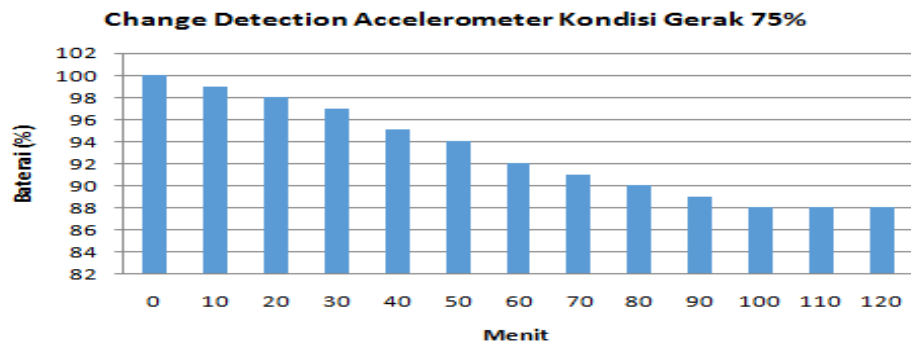
Gambar 4.13. Perbandingan penurunan daya baterai kondisi gerak 50% dengan IPS periodik

Analisa pengujian dalam kondisi gerak 50%, pada Gambar 4.11 penurunan level baterai selama 120 menit menunjukkan pada level 86%. Rata-rata penurunan daya baterai pada Gambar 4.11 selama 120 menit adalah 1,2%. Pengujian pada menit ke 100, 110 posisi baterai pada level yang sama. Analisa pada Gambar 4.12 penurunan level baterai selama 120 menit menunjukkan pada level 90%, rata-rata penurunan daya pada Gambar 4.12 selama 120 menit adalah 0,9%. Pengujian pada menit ke 40, 70, 80, 90, 100, 110, 120 posisi baterai pada level yang sama. Gambar 4.13 menunjukkan perbandingan pengujian dalam kondisi gerak 50% antara IPS scanning tanpa *change detection*, didapatkan penghematan energi baterai sebesar 1,538% dengan *change detection* menggunakan *Bluetooth* dan 3,615% dengan menggunakan *accelerometer*.

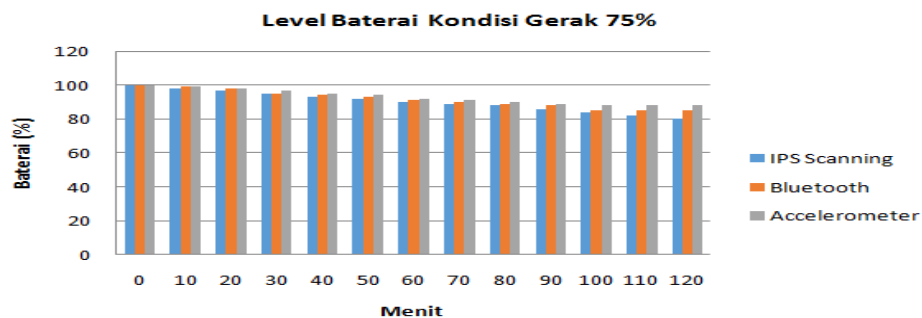
f. Hasil pengujian 75% dalam kondisi gerak



Gambar 4.14. Penurunan daya baterai kondisi gerak 75% dengan *Bluetooth*



Gambar 4.15. Penurunan daya baterai kondisi gerak 75% dengan *Accelerometer*



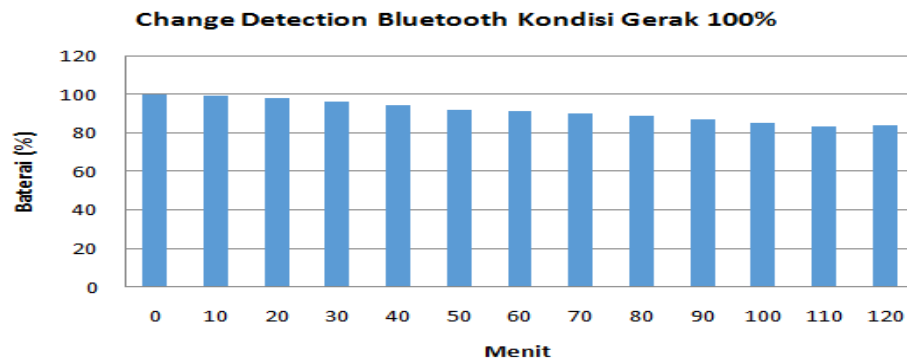
Gambar 4.16. Perbandingan penurunan daya baterai kondisi gerak 75% dengan IPS periodik

Analisa pengujian dalam kondisi gerak 75%, pada Gambar 4.14 penurunan level baterai selama 120 menit menunjukkan pada level 85%. Rata-rata penurunan daya baterai pada Gambar 4.14 selama 120 menit adalah 1,1%. Pengujian pada menit ke 100, 110, 120 posisi baterai pada level yang

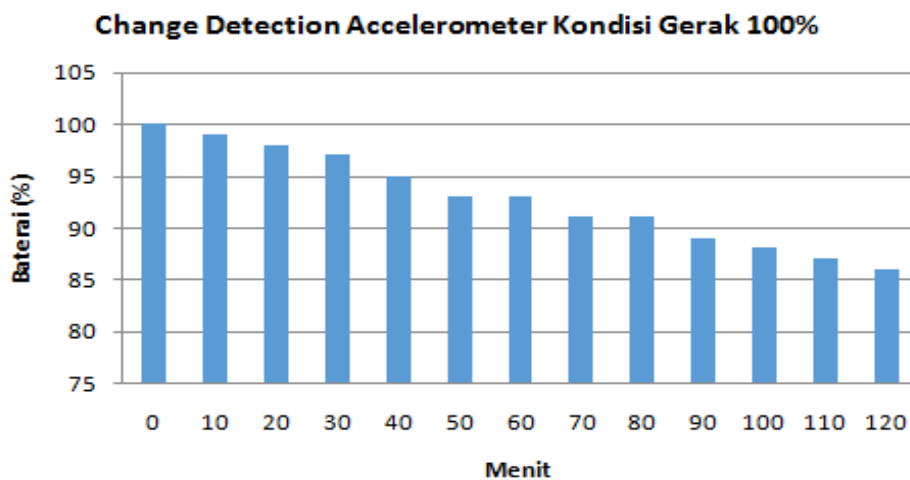


sama. Analisa pada Gambar 4.15 penurunan level baterai selama 120 menit menunjukkan pada level 88%, rata-rata penurunan daya pada Gambar 4.15 selama 120 menit adalah 1%. Pengujian pada menit ke 100, 110, 120 posisi baterai pada level yang sama. Gambar 4.16 menunjukkan perbandingan pengujian dalam kondisi gerak 75% antara IPS scanning tanpa *change detection*, didapatkan penghematan energi baterai sebesar 1,384% dengan *change detection* menggunakan *Bluetooth* dan 2,692% dengan menggunakan *accelerometer*.

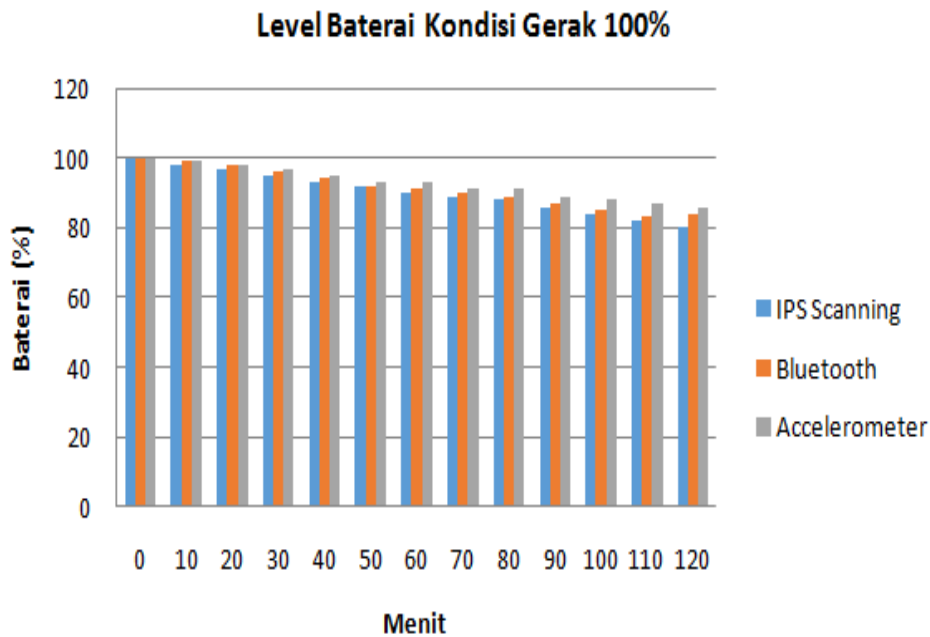
g. Hasil pengujian 100% dalam kondisi gerak



Gambar 4.17. Penurunan daya baterai kondisi gerak 100% dengan *Bluetooth*



Gambar 4.18. Penurunan daya baterai kondisi gerak 100% dengan *Accelerometer*



Gambar 4.19. Perbandingan penurunan daya baterai kondisi gerak 100% dengan IPS periodik

Analisa pengujian dalam kondisi gerak 100%, pada Gambar 4.17 penurunan level baterai selama 120 menit menunjukkan pada level 84%. Rata-rata penurunan daya baterai pada Gambar 4.17 selama 120 menit adalah 1,3%. Analisa pada Gambar 4.18 penurunan level baterai selama 120 menit menunjukkan pada level 86%, rata-rata penurunan daya pada Gambar 4.18 selama 120 menit adalah 1,2%. Pengujian pada menit ke 50, 60, 70, 80 posisi baterai pada level yang sama. Gambar 4.19 menunjukkan perbandingan pengujian dalam kondisi gerak 100% antara IPS scanning tanpa *change detection*, didapatkan penghematan energi baterai sebesar 1,076% dengan *change detection* menggunakan *Bluetooth* dan 2,538% dengan menggunakan *accelerometer*. Ringkasan dari hasil penggunaan daya baterai untuk pengujian dalam kondisi diam 100%, pengujian 10% gerak, pengujian 30% gerak, pengujian 50% gerak, pengujian 75% gerak dan pengujian 100% dapat dilihat pada Tabel 4.1.

h. Hasil pengujian berapa persen penghematan untuk efisiensi energi

Tabel 4.1 Ringkasan hasil penggunaan daya baterai

No	Skenario	Persentase Penghematan Energi Baterai	
		Bluetooth	Accelerometer
1	Pengujian dalam kondisi diam 100%	4,923	7,076
2	Pengujian 10% gerak	4,076	5,615
3	Pengujian 30% gerak	3	4,769
4	Pengujian 50% gerak	1,538	3,615
5	Pengujian 75% gerak	1,384	2,692
6	Pengujian 100% gerak	1,076	2,538
		2,666	4,384

Keterangan pada Tabel 4.1. menunjukkan penghematan daya baterai selama 120 menit dengan *change detection* kombinasi diam dan gerak didapatkan total rata-rata keseluruhan sebesar 2,666% dengan menggunakan Bluetooth dan 4,384% dengan menggunakan *accelerometer*.

## 2. Hasil Uji Coba Akurasi *Change Detection*

a) Akurasi *change detection* dengan Bluetooth

Tabel 4.2 Hasil uji coba akurasi *change detection* menggunakan Bluetooth dengan 10 data training dengan nilai *Threshold* 11

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	11	12	Gerak	-1	1	0,09
2	11	6	Gerak	5	5	0,45
3	11	11	Gerak	5	5	0,00
4	11	9	Gerak	2	2	0,18
5	11	14	Gerak	-3	3	0,27
6	11	8	Gerak	3	3	0,27
7	11	5	Gerak	6	6	0,55
8	11	13	Gerak	-2	2	0,18
9	11	9	Gerak	2	2	0,18
10	11	5	Gerak	6	6	0,55
					Sum Error	2,64
					Avg Error	0,26
					MAPE	26,36
					Akurasi	73,64

Keterangan Tabel 4.2 menunjukkan, untuk *threshold* 11 didapatkan akurasi sebesar 71 % dengan persentase error sebesar 29%.

Tabel 4.3 Hasil uji coba akurasi *change detection* menggunakan *Bluetooth* dengan 10 data training dengan nilai *Threshold* 8

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	8	6	Gerak	2	2	0,25
2	8	9	Gerak	-1	1	0,13
3	8	12	Gerak	-4	4	0,50
4	8	5	Gerak	3	3	0,38
5	8	10	Gerak	-2	2	0,25
6	8	6	Gerak	2	2	0,25
7	8	6	Gerak	2	2	0,25
8	8	8	Gerak	0	0	0,00
9	8	13	Gerak	-5	5	0,63
10	8	7	Gerak	1	1	0,13
					Sum Error	2,75
					Avg Error	0,28
					MAPE	27,5
					Akurasi	72,5

Keterangan Tabel 4.3 menunjukkan, untuk *threshold* 8 didapatkan akurasi sebesar 72,5 % dengan persentase error sebesar 27,5% untuk *change detection* menggunakan *Bluetooth*.

Tabel 4.4 Hasil uji coba akurasi *change detection* menggunakan *Bluetooth* dengan 10 data training dengan nilai *Threshold* 6

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	6	9	Gerak	-3	3	0,50
2	6	14	Gerak	-8	8	1,33
3	6	7	Gerak	-1	1	0,17
4	6	8	Gerak	-2	2	0,33
5	6	5	Gerak	1	1	0,17
6	6	12	Gerak	-6	6	1,00
7	6	6	Gerak	0	0	0,00
8	6	4	Gerak	2	2	0,33
9	6	10	Gerak	-4	4	0,67
10	6	7	Gerak	-1	1	0,17
					Sum Error	4,67
					Avg Error	0,47
					MAPE	46,67
					Akurasi	53,33

Keterangan Tabel 4.4 menunjukkan, untuk *threshold* 6 didapatkan akurasi sebesar 53,33 % dengan persentase error sebesar 46,67%.

Tabel 4.5 Hasil uji coba akurasi *change detection* menggunakan *Bluetooth* dengan 10 data training dengan nilai *Threshold* 10

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	10	9	Gerak	1	1	0,10
2	10	13	Gerak	-3	3	0,30
3	10	7	Gerak	3	3	0,30
4	10	8	Gerak	2	2	0,20
5	10	5	Gerak	5	5	0,50
6	10	9	Gerak	1	1	0,10
7	10	7	Gerak	3	3	0,30
8	10	5	Gerak	5	5	0,50
9	10	5	Gerak	5	5	0,50
10	10	11	Gerak	-1	1	0,10
					Sum Error	2,90
					Avg Error	0,29
					MAPE	29,00
					Akurasi	71,00

Keterangan Tabel 4.5 menunjukkan, untuk *threshold* 10 didapatkan akurasi sebesar 71 % dengan persentase error sebesar 29%.

Tabel 4.6 Ringkasan akurasi *change detection* dengan *Bluetooth*

Ringkasan Akurasi Bluetooth	
Threshold	Akurasi %
11	73,64
8	72,5
6	53,33
10	71

Keterangan Tabel 4.6 menunjukkan, untuk *threshold* 11 didapatkan akurasi sebesar 68,18 %, dengan *threshold* 8 didapatkan akurasi sebesar 72,5 %, dengan *threshold* 6 didapatkan akurasi sebesar 53,33 % dan *threshold* 10 didapatkan akurasi sebesar 71 %.

b) Akurasi *change detection* dengan Accelerometer

Tabel 4.7 Hasil uji coba akurasi *change detection* menggunakan Accelerometer dengan 10 data training dengan nilai *Threshold* 1,238

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	1,238	1,114	Gerak	0,1243	0,1243	0,1004
2	1,238	1,395	Gerak	-0,1569	0,1569	0,1268
3	1,238	0,783	Gerak	0,4551	0,4551	0,3676
4	1,238	1,499	Gerak	-0,2609	0,2609	0,2108
5	1,238	1,324	Gerak	-0,0859	0,0859	0,0694
6	1,238	1,148	Gerak	0,0901	0,0901	0,0727
7	1,238	1,293	Gerak	-0,0549	0,0549	0,0444
8	1,238	0,677	Gerak	0,5611	0,5611	0,4532
9	1,238	2,051	Gerak	-0,8129	0,8129	0,6566
10	1,238	1,586	Gerak	-0,3479	0,3479	0,2810
					Sum Error	2,3828
					Avg Error	0,2383
					MAPE	23,8282
					Akurasi	76,1718

Keterangan pada Tabel 4.7. menunjukkan, untuk *threshold* 1,238 didapatkan akurasi sebesar 76,17 % dengan persentase error sebesar 23,82%.

Tabel 4.8 Hasil uji coba akurasi *change detection* menggunakan *accelerometer* dengan 10 data training dengan nilai *Threshold* 0,818

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	0,818	1,191	Gerak	-0,3730	0,3730	0,4560
2	0,818	1,256	Gerak	-0,4380	0,4380	0,5355
3	0,818	0,544	Gerak	0,2740	0,2740	0,3350
4	0,818	0,825	Gerak	-0,0070	0,0070	0,0086
5	0,818	1,030	Gerak	-0,2120	0,2120	0,2592
6	0,818	0,698	Gerak	0,1200	0,1200	0,1467
7	0,818	0,498	Gerak	0,3200	0,3200	0,3912
8	0,818	0,483	Gerak	0,3350	0,3350	0,4095
9	0,818	1,274	Gerak	-0,4560	0,4560	0,5575
10	0,818	0,613	Gerak	0,2050	0,2050	0,2506
					Sum Error	3,3496
					Avg Error	0,3350
					MAPE	33,4963
					Akurasi	66,5037

Keterangan pada Tabel 4.8. menunjukkan akurasi *change detection* selama 10 kali pengujian dengan kondisi bergerak jalan untuk *accelerometer*. Hasilnya didapatkan 66,50 % untuk akurasi dan persentase error sebesar 33,49%.

Tabel 4.9 Hasil uji coba akurasi *change detection* menggunakan *accelerometer* dengan 10 data training dengan nilai *Threshold* 0,519

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	0,519	0,613	Gerak	-0,0940	0,0940	0,1811
2	0,519	0,393	Gerak	0,1260	0,1260	0,2428
3	0,519	1,037	Gerak	-0,5180	0,5180	0,9981
4	0,519	0,407	Gerak	0,1120	0,1120	0,2158
5	0,519	0,601	Gerak	-0,0820	0,0820	0,1580
6	0,519	1,130	Gerak	-0,6110	0,6110	1,1773
7	0,519	0,309	Gerak	0,2100	0,2100	0,4046
8	0,519	1,101	Gerak	-0,5820	0,5820	1,1214
9	0,519	0,958	Gerak	-0,4390	0,4390	0,8459
10	0,519	0,282	Gerak	0,2370	0,2370	0,4566
					Sum Error	5,8015
					Avg Error	0,5802
					MAPE	58,0154
					Akurasi	41,9846

Keterangan pada Tabel 4.9. menunjukkan, untuk *threshold* 0,519 didapatkan akurasi sebesar 41,98 % dengan persentase error sebesar 58,01%.

Tabel 4.10 Hasil uji coba akurasi *change detection* menggunakan *accelerometer* dengan 10 data training dengan nilai *Threshold* 1,117

Uji Coba Ke	Threshold	Nilai Sensor	Hasil	Nilai Error	Abs Error	% error
1	1,117	1,576	Gerak	-0,4590	0,4590	0,4109
2	1,117	0,417	Gerak	0,7000	0,7000	0,6267
3	1,117	0,943	Gerak	0,1740	0,1740	0,1558
4	1,117	1,652	Gerak	-0,5350	0,5350	0,4790
5	1,117	2,339	Gerak	-1,2220	1,2220	1,0940
6	1,117	0,177	Gerak	0,9400	0,9400	0,8415
7	1,117	0,473	Gerak	0,6440	0,6440	0,5765
8	1,117	0,677	Gerak	0,4400	0,4400	0,3939
9	1,117	0,830	Gerak	0,2870	0,2870	0,2569
10	1,117	1,054	Gerak	0,0630	0,0630	0,0564
					Sum Error	4,8917
					Avg Error	0,4892
					MAPE	48,9167
					Akurasi	51,0833

Keterangan pada Tabel 4.10. menunjukkan, untuk *threshold* 1,117 didapatkan akurasi sebesar 51,08 % dengan persentase error sebesar 48,91%.

Tabel 4.11 Ringkasan akurasi *change detection* dengan *accelerometer*

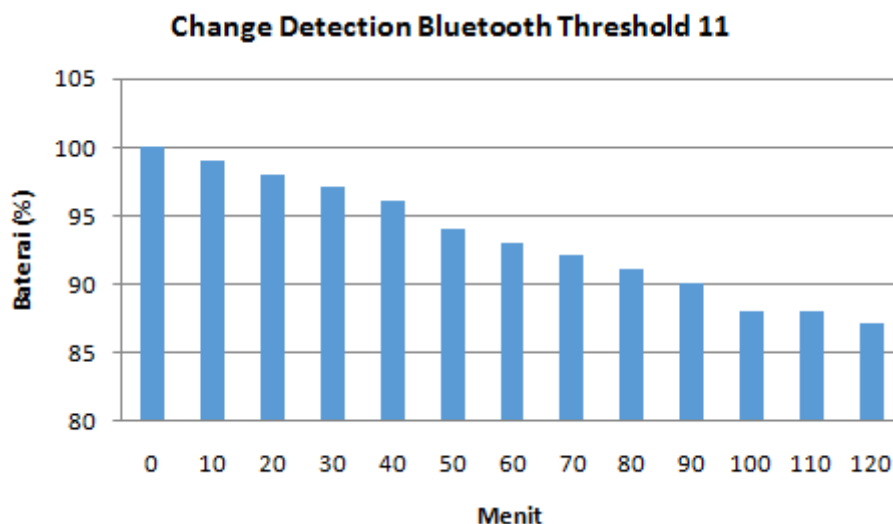
Ringkasan Akurasi Accelerometer	
Threshold	Akurasi %
1,238	76,17
0,818	66,5
0,519	41,98
1,117	51,08

Keterangan Tabel 4.11 menunjukkan, untuk *threshold* 1,238 didapatkan akurasi sebesar 76,17 %, dengan *threshold* 0,818 didapatkan akurasi sebesar 66,5 %, dengan *threshold* 0,519 didapatkan akurasi sebesar 41,98 % dan *threshold* 1,117 didapatkan akurasi sebesar 51,08 %.

c) Pengaruh akurasi terhadap penghematan baterai

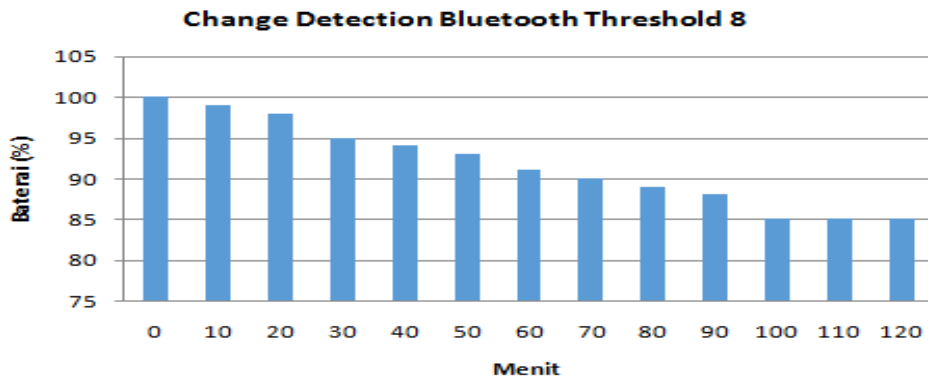
Berikut hasil pengujian akurasi berdasarkan *threshold* terhadap penurunan daya baterai.

a. *Change Detection* dengan *Bluetooth*

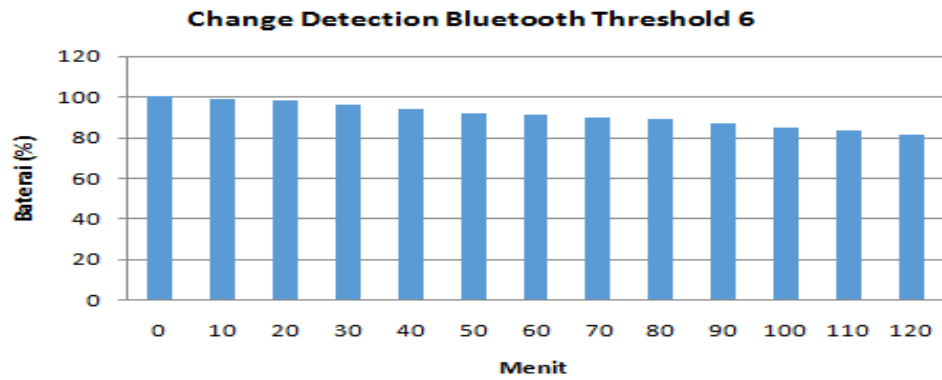


Gambar 4.20. Penurunan daya baterai dengan *change detection* menggunakan *Bluetooth threshold 11*

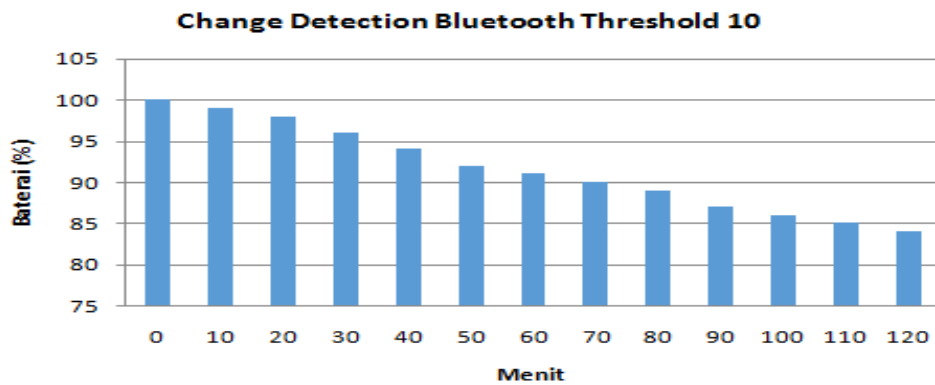




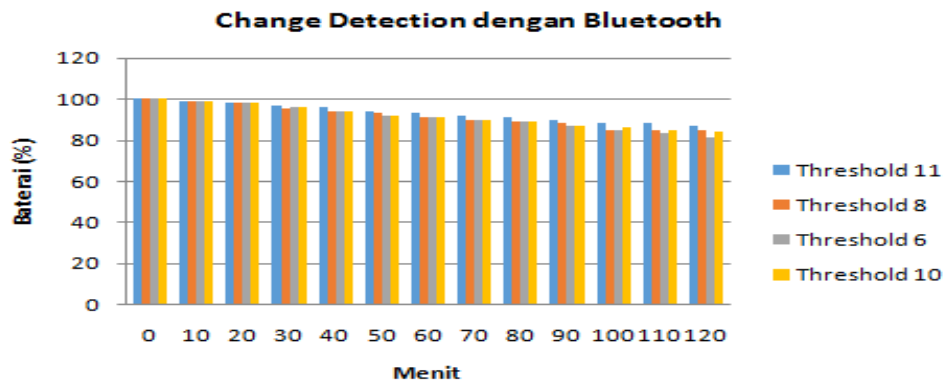
Gambar 4.21. Penurunan daya baterai dengan *change detection* menggunakan *Bluetooth threshold 8*



Gambar 4.22. Penurunan daya baterai dengan *change detection* menggunakan *Bluetooth threshold 6*



Gambar 4.23. Penurunan daya baterai dengan *change detection* menggunakan *Bluetooth threshold 10*



Gambar 4.24. Perbandingan penurunan daya baterai dengan *threshold* pada *change detection* menggunakan *Bluetooth*

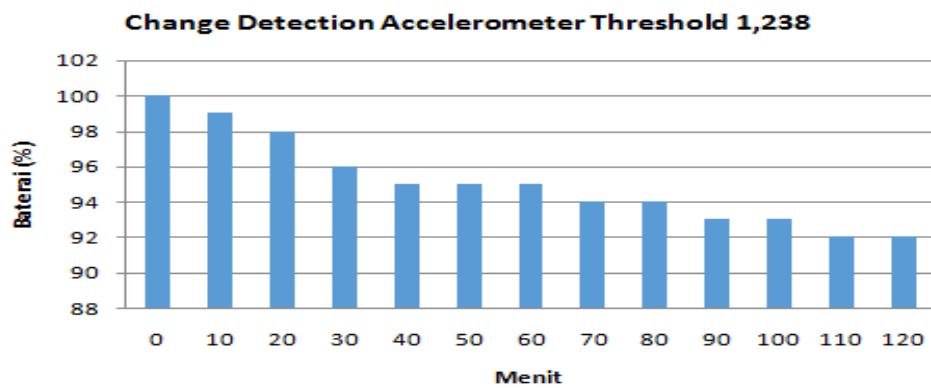
Analisa pengujian *change detection* menggunakan *Bluetooth* dengan *threshold* 11, pada Gambar 4.20 penurunan *level* baterai selama 120 menit menunjukkan pada level 87%. Pada Gambar 4.21 penurunan *level* baterai selama 120 menit menunjukkan pada *level* 85%, pada Gambar 2.22 penurunan *level* baterai selama 120 menit menunjukkan pada *level* 81% dan pada Gambar 2.23 penurunan *level* baterai selama 120 menit menunjukkan pada level 84%. Gambar 2.24 merupakan grafik perbandingan antara *change detection* menggunakan *Bluetooth* dengan *threshold* 11, 8, 6 dan 10. Ringkasan penghematan baterai dengan *change detection* menggunakan *accelerometer* dengan *threshold* 11, 8, 6 dan 10 dapat dilihat pada Tabel 4.12.

Tabel 4.12 Ringkasan pengaruh akurasi *change detection* dengan *Bluetooth* terhadap penghematan baterai

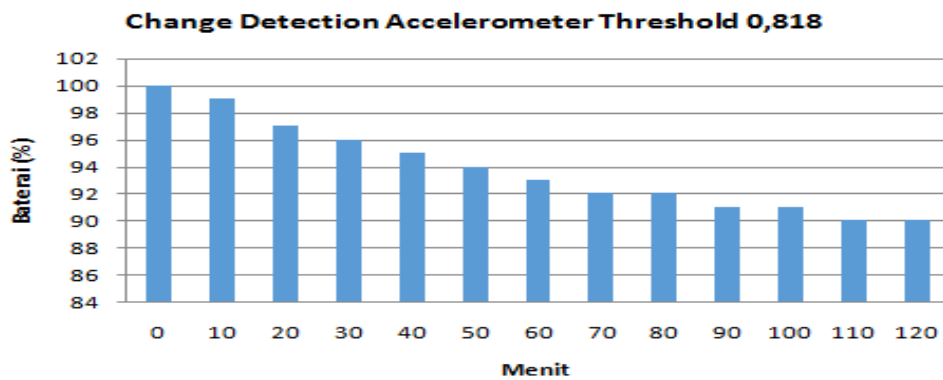
No	Skenario	Persentase Penghematan Energi Baterai (%)	Akurasi Change Detection (%)
		Bluetooth	
1	Pengujian dengan threshold 11	3	68,180
2	Pengujian dengan threshold 8	1,384	72,500
3	Pengujian dengan threshold 6	0,846	53,330
4	Pengujian dengan threshold 10	1,307	71,000
		1,634	66,253

Keterangan Tabel 4.12 menunjukkan, untuk *change detection* menggunakan *Bluetooth* dengan *threshold* 11 didapatkan penghematan energi sebesar 3%, dengan *threshold* 8 didapatkan penghematan energi sebesar 1,384%, dengan *threshold* 6 didapatkan penghematan energi sebesar 0,846% dan dengan *threshold* 10 didapatkan penghematan energi sebesar 1,634%.

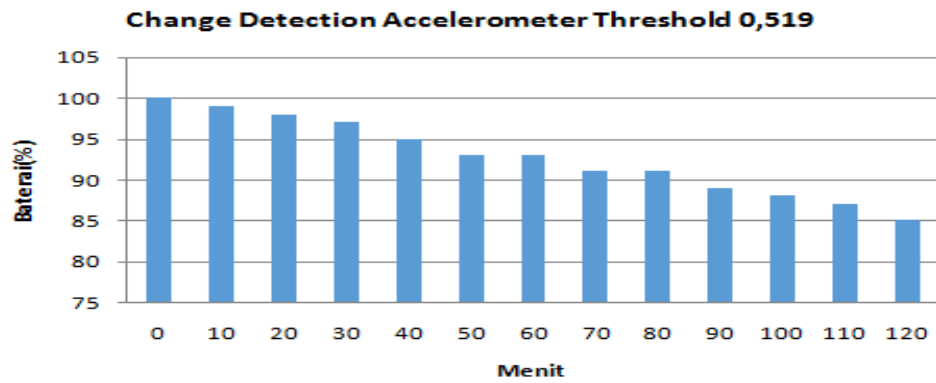
b. *Change Detection* dengan *Accelerometer*



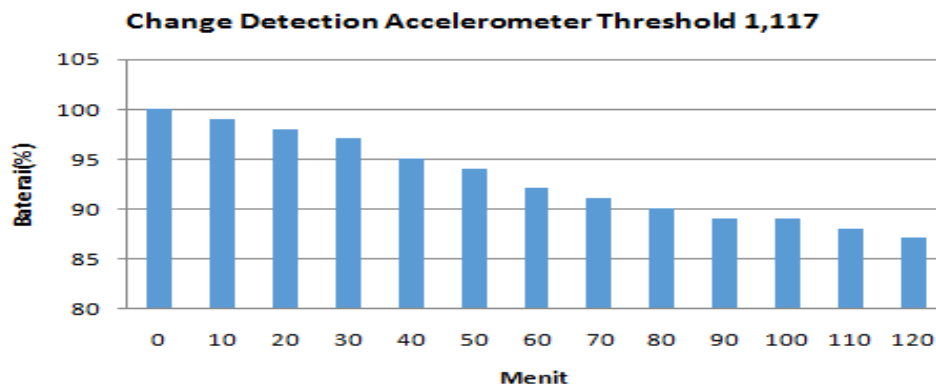
Gambar 4.25. Penurunan daya baterai dengan *change detection* menggunakan *accelerometer threshold* 1,238



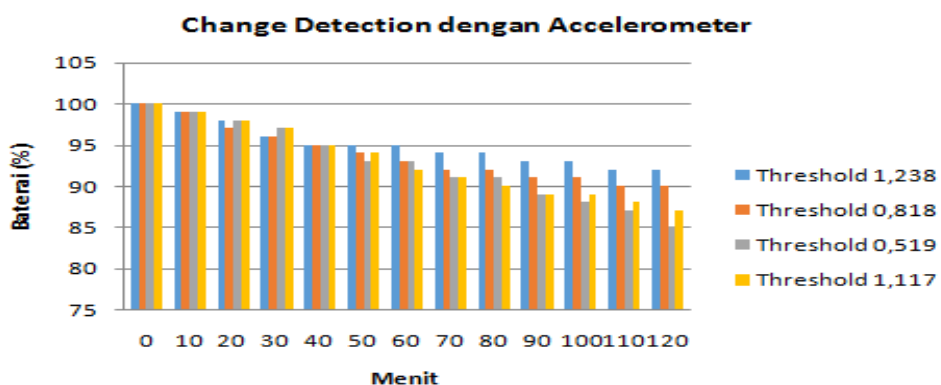
Gambar 4.26. Penurunan daya baterai dengan *change detection* menggunakan *accelerometer threshold* 0,818



Gambar 4.27. Penurunan daya baterai dengan *change detection* menggunakan *accelerometer threshold 0,519*



Gambar 4.28. Penurunan daya baterai dengan *change detection* menggunakan *accelerometer threshold 1,117*



Gambar 4.29. Perbandingan penurunan daya baterai dengan *threshold* pada *change detection* menggunakan *accelerometer*

Analisa pengujian *change detection* menggunakan *accelerometer* dengan *threshold* 1,238, pada Gambar 4.25 penurunan *level* baterai selama 120 menit menunjukkan pada level 92%. Pada Gambar 4.26 penurunan *level* baterai selama 120 menit menunjukkan pada *level* 90%, pada Gambar 2.27 penurunan *level* baterai selama 120 menit menunjukkan pada *level* 85% dan pada Gambar 2.28 penurunan *level* baterai selama 120 menit menunjukkan pada level 87%. Gambar 2.29 merupakan grafik perbandingan antara *change detection* menggunakan *accelerometer* dengan *threshold* 1,238, 0,818, 0,519 dan 1,117. Ringkasan penghematan baterai dengan *change detection* menggunakan *accelerometer* dengan *threshold* 1,238, 0,818, 0,519 dan 1,117 dapat dilihat pada Tabel 4.13.

Tabel 4.13 Ringkasan pengaruh akurasi *change detection* dengan *accelerometer* terhadap penghematan baterai

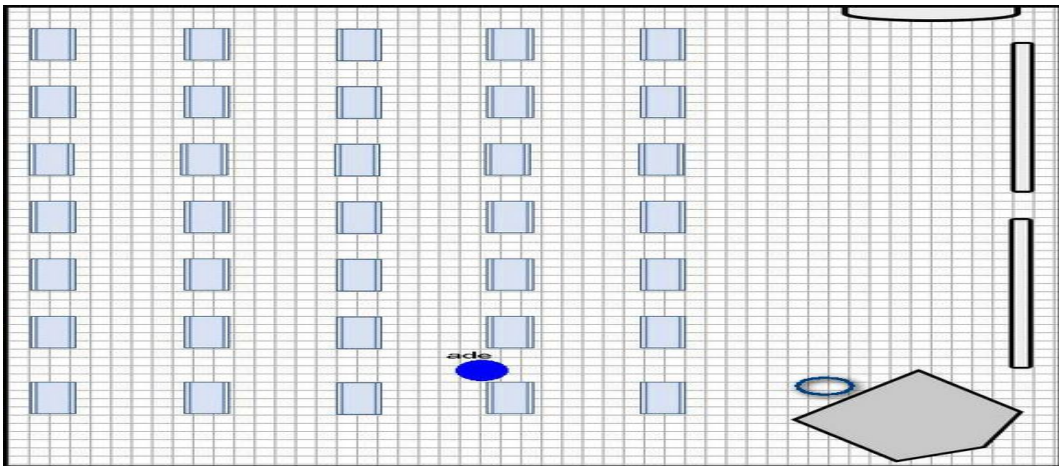
No	Skenario	Persentase Penghematan Energi Baterai (%)	Akurasi Change Detection (%)
		Accelerometer	
1	Pengujian dengan threshold 1,238	4,769	76,17
2	Pengujian dengan threshold 0,818	3,538	66,5
3	Pengujian dengan threshold 0,519	2,461	41,98
4	Pengujian dengan threshold 1,117	2,692	51,08
		3,365	58,933

Keterangan Tabel 4.13 menunjukkan, untuk *change detection* menggunakan *accelerometer* dengan *threshold* 1,238 didapatkan penghematan energi sebesar 4,769%, dengan *threshold* 0,818 didapatkan penghematan energi sebesar 3,538%, dengan *threshold* 0,519 didapatkan penghematan energi sebesar 2,461% dan dengan *threshold* 1,117 didapatkan penghematan energi sebesar 2,692%.

### 3. Hasil Uji Coba Akurasi Lokasi

Hasil pengujian lokasi berdasarkan reporting yang dikirim ke server, dibandingkan dengan jarak fisik, untuk mengetahui jarak yang sebenarnya. Uji coba dilakukan pad lokasi yang memiliki POF yang memiliki ID 10 dengan koordinat  $x = 400$  dan  $y = 400$ , kemudian menjalankan training sebanyak 10 data

dengan interval 10 detik. Gambar 4.20 menunjukkan *map* posisi *user* pada lokasi di titik POF dengan ID 10 dan lingkaran bulat berwarna hitam menunjukkan posisi fisik pengujian.



Gambar 4.30. Titik yang digunakan uji coba jarak fisik

Berdasarkan pengukuran yang dilakukan secara real, didapatkan hasil uji coba seperti pada tabel di bawah berikut.

Tabel 4.14 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ) dengan IPS *scanning* tanpa *change detection*

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	210	220	2,100
3	270	106,667	2,800
4	390	220	2,000
5	210	50	3,050
6	0	0	0,000
7	300	570	0,700
8	210	220	2,100
9	210	60	2,400
10	300	135	2,250
Rata-rata			1,840

Keterangan pada Tabel 4.14. menjelaskan bahwa lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ), setelah dilakukan training selama 10 kali menghasilkan jarak secara fisik sebesar 1,840 meter dengan IPS *scanning* periodik dengan delay 3 detik.

a) Akurasi lokasi dengan *change detection* menggunakan *Bluetooth*

Tabel 4.15 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan *change detection* menggunakan *Bluetooth threshold 11*

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	300	570	0,700
3	210	135	2,070
4	210	220	2,100
5	300	570	0,700
6	300	570	0,700
7	30	570	2
8	300	570	0,700
9	120	50	3,150
10	210	135	2,100
Rata-rata			1,522

Keterangan pada Tabel 4.15. menjelaskan pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,522 meter dengan IPS *change detection* menggunakan *Bluetooth* dengan *threshold 11*.

Tabel 4.16 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan *change detection* menggunakan *Bluetooth threshold 8*

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	120	50	3,150
3	210	135	2,070
4	210	310	1,150
5	50	200	0,650
6	300	570	0,700
7	0	0	0
8	300	570	0,700
9	200	50	2,700
10	50	200	0,650
Rata-rata			1,277

Keterangan pada Tabel 4.16. menjelaskan untuk lokasi titik POF ID 10 ( $x = 350, y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,277 meter dengan IPS *change detection* menggunakan *Bluetooth* dengan *threshold* 8.

Tabel 4.17 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan *change detection* menggunakan *Bluetooth threshold* 6

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	30	220	1,900
3	300	570	0,700
4	50	200	0,650
5	255	300	1,200
6	210	310	1,150
7	210	310	1,150
8	220	340	0,700
9	250	200	2,800
10	30	200	0,850
Rata-rata			1,210

Keterangan pada Tabel 4.17. menjelaskan untuk lokasi titik POF ID 10 ( $x = 350, y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,210 meter dengan IPS *change detection* menggunakan *Bluetooth* dengan *threshold* 6.

Tabel 4.18 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350, y = 350$ ) IPS dengan *change detection* menggunakan *Bluetooth threshold* 10

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	570	40	1,200
3	210	135	2,070
4	210	310	1,150
5	240	106	2,150
6	300	570	0,700
7	30	22,35	1
8	300	570	0,700
9	200	50	2,700
10	255	440	0,900
Rata-rata			1,337



Keterangan pada Tabel 4.18. menjelaskan untuk lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,277 meter dengan IPS *change detection* menggunakan *Bluetooth* dengan *threshold* 10. Ringkasan akurasi posisi dengan *change detection* menggunakan *Bluetooth* dapat dilihat pada Tabel 4.19.

Tabel 4.19 Ringkasan akurasi posisi dengan *change detection* menggunakan *Bluetooth*

No	Skenario	Persentase akurasi posisi
		Bluetooth
1	Pengujian akurasi dengan threshold 11	1,552
2	Pengujian akurasi dengan threshold 8	1,277
3	Pengujian akurasi dengan threshold 6	1,210
4	Pengujian akurasi dengan threshold 10	1,337
		1,344

Keterangan Tabel 4.19 menunjukkan, untuk akurasi posisi dengan *change detection* menggunakan *Bluetooth* dengan *threshold* 11 didapatkan jarak fisik sebesar 1,552 meter, *change detection* menggunakan *Bluetooth* dengan *threshold* 8 didapatkan jarak fisik sebesar 1,277 meter, *change detection* menggunakan *accelerometer* dengan *threshold* 6 didapatkan jarak fisik sebesar 1,210 meter dan *change detection* menggunakan *accelerometer* dengan *threshold* 10 didapatkan jarak fisik sebesar 1,337 meter. Hasil ringkasan dari ke empat pengujian akurasi didapatkan jarak secara fisik rata-rata sebesar 1,344 meter.

b) Akurasi lokasi dengan *change detection* menggunakan *accelerometer*

Tabel 4.20 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ) IPS dengan *change detection* menggunakan *accelerometer threshold* 1,238

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	300	570	0,700
3	0	0	0
4	210	135	2,100
5	255	440	0,900
6	240	106,67	2,850
7	300	570	0,700
8	210	310	1,150
9	210	135	2
10	300	570	0,700
Rata-rata			1,220

Keterangan pada Tabel 4.20. menjelaskan pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,220 meter dengan IPS *change detection* menggunakan *accelerometer* dengan *threshold* 1,238.

Tabel 4.21 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ) IPS dengan *change detection* menggunakan *accelerometer threshold* 0,818

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	240	106,67	2,850
3	210	310	1,150
4	210	135	2,100
5	0	0	0,000
6	210	310	1,150
7	30	220	1,900
8	300	500	0,900
9	0	0	0,000
10	50	200	0,650
Rata-rata			1,170

Keterangan pada Tabel 4.21 menjelaskan pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,170 meter dengan IPS *change detection* menggunakan *accelerometer* dengan *threshold* 0,818.

Tabel 4.22 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ) IPS dengan *change detection* menggunakan *accelerometer threshold 0,519*

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	210	310	1,150
3	120	50	3,150
4	300	220	1,250
5	255	300	1,200
6	210	310	1,150
7	210	310	1,150
8	0	0	0,000
9	300	500	0,900
10	0	0	0,000
Rata-			1,095

Keterangan pada Tabel 4.22 menjelaskan pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,095 meter dengan IPS *change detection* menggunakan *accelerometer* dengan *threshold 0,519*.

Tabel 4.23 Hasil uji coba akurasi posisi dengan 10 data training pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ) IPS dengan *change detection* menggunakan *accelerometer threshold 1,117*

Uji Coba Ke	x Pada Map	y Pada Map	Jarak Fisik (Meter)
1	350	350	1
2	0	0	0,000
3	30	220	1,900
4	300	500	0,900
5	0	0	0,000
6	210	135	2,100
7	210	310	1,150
8	210	310	1,150
9	200	50	2,700
10	210	310	1,150
Rata-rata			1,205

Keterangan pada Tabel 4.23 menjelaskan pada lokasi titik POF ID 10 ( $x = 350$ ,  $y = 350$ ), setelah dilakukan training selama 10 kali selama interval 10 detik menghasilkan jarak secara fisik sebesar 1,205 meter dengan IPS *change detection*

menggunakan *accelerometer* dengan *threshold* 1,117. Ringkasan akurasi posisi dengan *change detection* menggunakan *accelerometer* dapat dilihat pada Tabel 4.24.

Tabel 4.24 Ringkasan akurasi posisi dengan *change detection* menggunakan *accelerometer*

No	Skenario	Persentase akurasi posisi
		Accelerometer
1	Pengujian akurasi dengan threshold 1,238	1,220
2	Pengujian akurasi dengan threshold 0,818	1,170
3	Pengujian akurasi dengan threshold 0,519	1,095
4	Pengujian akurasi dengan threshold 1,117	1,205
		1,173

Keterangan Tabel 4.24 menunjukkan, untuk akurasi posisi dengan *change detection* menggunakan *accelerometer* dengan *threshold* 1,238 didapatkan jarak fisik sebesar 1,220 meter, *change detection* menggunakan *accelerometer* dengan *threshold* 0,818 didapatkan jarak fisik sebesar 1,170 meter, *change detection* menggunakan *accelerometer* dengan *threshold* 0,519 didapatkan jarak fisik sebesar 1,095 meter dan *change detection* menggunakan *accelerometer* dengan *threshold* 1,117 didapatkan jarak fisik sebesar 1,205 meter. Hasil ringkasan dari ke empat pengujian akurasi didapatkan jarak secara fisik rata-rata sebesar 1,173 meter.

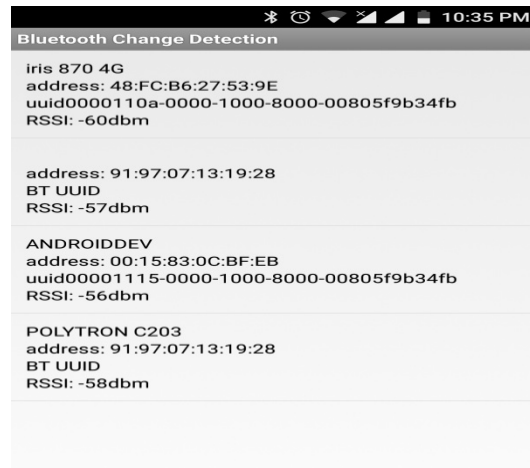
### 4.3.2. Hasil Uji Coba Fungsionalitas

Uji coba fungsionalitas dilakukan untuk mengetahui bahwa sistem berjalan sesuai *role* yang didesain. Beberapa uji coba yang dilakukan seperti berikut, scanning sinyal *Bluetooth*, respon *sensor accelerometer*, scanning sinyal WiFi, hasil WiFi *scanning*.

#### 1. Scanning Sinyal Bluetooth

Fungsi dari *scanning* kekuatan sinyal Bluetooth ini adalah menampilkan daftar Bluetooth yang sudah didefinisikan terlebih dahulu *mac address* nya

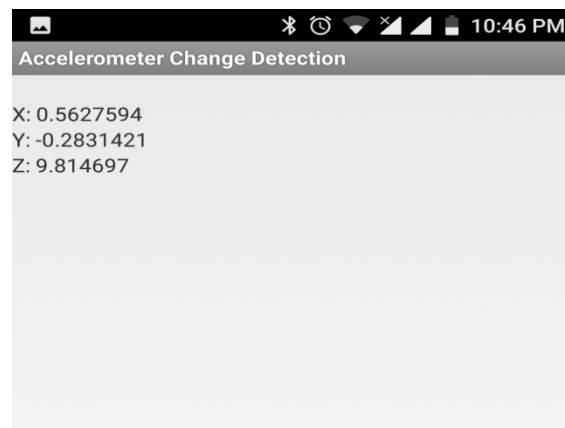
dengan tujuan untuk kebutuhan *sampling* yang akan diolah untuk proses penentuan *silent zone*. Informasi nya bisa dilihat seperti pada Gambar 4.20. di bawah berikut.



Gambar 4.31. *Scanning* sinyal Bluetooth

## 2. Respon Sensor Accelerometer

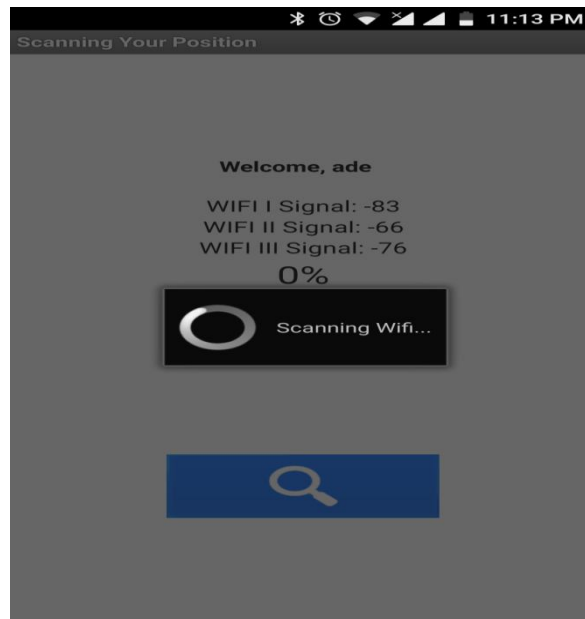
Fungsi dari *accelerometer* ini tujuannya sama dengan *scanning* sinyal *Bluetooth*, dengan tujuan untuk kebutuhan *sampling* yang akan diolah untuk proses penentuan *silent zone* agar bisa mengidentifikasi *user* apakah sedang bergerak atau diam sebelum melakukan proses *scanning* sinyal WiFi. Informasi nya bisa dilihat seperti pada Gambar 4.32. di bawah berikut.



Gambar 4.32. Respon Sensor *Accelerometer*

### 3. Scanning Sinyal WiFi

Proses scanning sinyal WiFi akan berjalan secara otomatis ketika *user* sedang bergerak, pada proses awal *user* terlebih dahulu melakukan *login* ke sistem. Fungsi ini dilakukan untuk memperoleh data berupa kekuatan sinyal dari setiap Wi-Fi yang telah ditentukan. Informasinya bisa dilihat seperti pada Gambar 4.33. di bawah berikut.



Gambar 4.33. Scanning Sinyal WiFi

### 4. Hasil WiFi Scanning

Proses mengumpulkan informasi raw data dari WiFi scanning sinyal WiFi. Setiap WiFi scanning berisi nilai RSSI instan dari masing-masing AP kemudian dikirim ke server. Informasi hasil bisa dilihat seperti pada Gambar 4.23. di bawah berikut.

id_fs	id_finger	ss1	ss2	ss3
2001	0	-86	-65	-93
2002	0	-85	-77	-94
2003	0	-85	-77	-94
2004	0	-85	-77	-94
2005	0	-85	-77	-94
2006	0	-85	-77	-94
2007	0	-90	-79	-95
2008	0	-89	-81	-95
2009	0	-90	-68	-95
2010	0	-85	-68	-93

Gambar 4.34. Hasil WiFi Scanning

Keterangan pada Gambar 4.34. menjelaskan untuk lokasi kolom ss1 merupakan kolom untuk menyimpan informasi kekuatan sinyal WiFi I, kolom ss2 untuk menyimpan informasi kekuatan sinyal WiFi II, kolom ss3 untuk menyimpan informasi kekuatan sinyal WiFi III, proses penyimpanan sampling dilakukan setelah proses *scanning* selesai.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Pada bab ini akan dibahas mengenai kesimpulan bagaimana mekanisme *change detection* untuk *scanning* sampling terhadap kekuatan sinyal secara adaptif untuk penghematan energi pada WiFi, yang dapat diambil dari sejumlah rangkaian penelitian yang telah dilakukan dan saran yang diambil untuk pengembangan penelitian.

#### **5.1. Kesimpulan**

Pengujian dan analisis terhadap *change detection* yang telah dilakukan menghasilkan beberapa kesimpulan penelitian, yaitu sebagai berikut:

a) Mekanisme *change detection* secara adaptif dilakukan dengan cara training raw data dengan mengambil nilai selisih perubahan percepatannya pada fluktuasi pada masing-masing nilai raw data, dari nilai selisih tersebut ditentukan nilai absolute nya. Kemudian dari deretan nilai absolute tersebut diambil nilai maksimal yang akan dijadikan sebagai nilai threshold atau ambang batas nilai untuk Silent Zone, dipilih nilai selisih yang paling besar dari semua training selama beberapa kali percobaan. Hal tersebut adalah untuk mengatur proses scanning yaitu *change detection*.

b) Pengaruh akurasi *change detection* dihasilkan beberapa kesimpulan seperti berikut :

1. Pengujian penghematan energi baterai dalam kondisi diam 100% dalam interval waktu selama 120 menit dengan interval 10 menit pengujian dihasilkan persentase penghematan daya baterai dengan *change detection* menggunakan *Bluetooth* sebesar 4,923% dan *accelerometer* sebesar 7,076% dibandingkan dengan IPS tanpa *Change detection*.
2. Pengujian penghematan energi baterai dengan kondisi gerak 10% dalam interval waktu selama 120 menit dengan interval 10 menit, dihasilkan persentase penghematan daya dengan *Bluetooth* sebesar 4,076% dan



*accelerometer* sebesar 5,615% dibandingkan dengan IPS tanpa *Change detection*.

3. Pengujian penghematan energi baterai dengan kondisi gerak 30% dalam interval waktu selama 120 menit dengan interval 10 menit, dihasilkan persentase penghematan daya dengan *Bluetooth* dan *accelerometer* masing-masing sebesar 3% dan 4,769% dibandingkan dengan IPS tanpa *Change detection*.
4. Pengujian penghematan energi baterai dengan kondisi gerak 50%, dihasilkan persentase penghematan daya dengan *Bluetooth* dan *accelerometer* masing-masing sebesar 1,538% dan 3,615% dibandingkan dengan IPS tanpa *Change detection*.
5. Pengujian penghematan energi baterai dengan kondisi gerak 75%, dihasilkan persentase penghematan daya dengan *Bluetooth* dan *accelerometer* masing-masing sebesar 1,384% dan 2,692% dibandingkan dengan IPS tanpa *Change detection*.
6. Pengujian penghematan energi baterai dengan kondisi gerak 100%, dihasilkan persentase penghematan daya dengan *Bluetooth* dan *accelerometer* masing-masing sebesar 1,076% dan 2,538% dibandingkan dengan IPS tanpa *Change detection*.

Dari keseluruhan pengujian untuk penghematan energi baterai dengan *change detection* dihasilkan nilai rata-rata untuk *Bluetooth* sebesar 2,666% dan *accelerometer* sebesar 4,384% dibandingkan dengan IPS tanpa *Change detection*.

7. Pengujian untuk akurasi *change detection* menggunakan *Bluetooth* dengan *threshold* 11 didapatkan akurasi sebesar 73,64 %, dengan *threshold* 8 didapatkan akurasi sebesar 72,5 %, dengan *threshold* 6 didapatkan akurasi sebesar 53,33 % dan *threshold* 10 didapatkan akurasi sebesar 71 %. *Change detection* dengan *accelerometer* dengan *threshold* 1,238 didapatkan akurasi sebesar 76,17 %, dengan *threshold* 0,818 didapatkan akurasi sebesar 66,5 %, dengan *threshold*

0,519 didapatkan akurasi sebesar 41,98 % dan *threshold* 1,117 didapatkan akurasi sebesar 51,08 %.

8. Pengaruh akurasi *change detection* terhadap penghematan energi dapat disimpulkan bahwa semakin tinggi akurasi *change detection* dalam penghematan energi lebih baik, tetapi pengaruh pada fungsi *change detection* kurang efisien.

9. Pengujian untuk akurasi lokasi disimpulkan bahwa *change detection* dengan *threshold* lebih besar akan menghasilkan akurasi posisi secara fisik lebih besar dibandingkan dengan *change detection* dengan *threshold* lebih kecil menghasilkan akurasi yang mendekati secara fisik dan di map. Seperti pada *threshold* 11 pada *Bluetooth* merupakan *threshold* terbesar, menghasilkan jarak fisik 1,552 meter dan *threshold* 6 merupakan nilai terkecil *threshold* pada *Bluetooth* menghasilkan jarak fisik 1,210. Untuk *change detection* menggunakan *accelerometer* dengan *threshold* terbesar yaitu 1,238 mendapatkan jarak fisik sebesar 1,220 meter dan *threshold* terkecil 0,519 didapatkan jarak fisik sebesar 1,095.

## 5.2. Saran

Pada sistem pengembangan mekanisme *change detection* ini memiliki kekurangan mengenai masalah responsif, terutama menggunakan *Bluetooth* ketika untuk mengidentifikasi *user* dalam kondisi bergerak dari posisi diam, pada *Bluetooth* harus menunggu selesai terlebih dahulu proses *sampling* untuk menentukan nilai *silent zone*. Karena pada *Bluetooth* ketika menyimpan satu data ada jeda beberapa detik pada satu data berikutnya. Hal ini perlu dilakukan penelitian lebih lanjut agar bisa responsif seperti pada *accelerometer*.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- Potgantwar, Amol, Jaspreet Kaur Kohli, and Prajakta Shirke. 2015. "Adaptive Rss Based Indoor Positioning System Using Rfid and Wireless." 4(4).
- Masoum, Alireza, Nirvana Meratnia, and Paul J.M. Havinga. 2013. "An Energy-Efficient Adaptive Sampling Scheme for Wireless Sensor Networks." *Proceedings of the 2013 IEEE 8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing: Sensing the Future, ISSNIP 2013* 1: 231–36.
- Kuxdorf-Alkirata, Nizam, Oliver Spathmann, Oliver Koch, and Dieter Brückmann. 2018. "Improved Energy Efficiency of Indoor Positioning Systems by Adaptive Sampling and Smart Post-Processing of Sensor Data." *2018 16th IEEE International New Circuits and Systems Conference, NEWCAS 2018*: 225–28.
- Gätej, Radu Călin, Jakob Eg Larsen, and Piotr Sapieżyński. 2013. "An Adaptive Approach to Mobile Sampling."
- Chen, Yi-Chao et al. 2005. "Sensor-Assisted Wi-Fi Indoor Location System for Adapting to Environmental Dynamics." : 118.
- Tuta, Jure, and Matjaz B. Juric. 2016. "A Self-Adaptive Model-Based Wi-Fi Indoor Localization Method." *Sensors (Switzerland)* 16(12).
- Niu, J. (2013). Energy-Efficient WiFi Fingerprint-Based Indoor Localization, (April), 1–28.
- Niu, Jianwei, Bowei Wang, Lei Shu, Trung Q. Duong, and Yuanfang Chen. 2015. "ZIL: An Energy-Efficient Indoor Localization System Using ZigBee Radio to Detect WiFi Fingerprints." *IEEE Journal on Selected Areas in Communications* 33(7):1431–42.
- Jekabsons, Gints, Vadim Kairish, and Vadim Zuravlyov. 2011. "An Analysis of Wi-Fi Based Indoor Positioning Accuracy." *Scientific Journal of Riga Technical University. Computer Sciences* 44(1):131–37.
- Liu, Hui, Houshang Darabi, Pat Banerjee, and Jing Liu. 2007. "Survey of Wireless Indoor Positioning Techniques and Systems." *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* 37(6):1067–80.
- Ma, Jun, Xuansong Li, Xianping Tao, and Jian Lu. 2008. "Cluster Filtered KNN: A WLAN-Based Indoor Positioning Scheme." *2008 IEEE International*

*Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM2008.*

- Xiang, Zhe. 2004. "A wireless LAN-based indoor positioning technology." *IBM Journal of Research and Development* 48(5.6):617–626.
- Feeney, L. M. and M. Nilsson. 2001. "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment." 1548–57.
- Balasubramanian, Niranjana, Aruna Balasubramanian, and Arun Venkataramani. 2009. "Energy Consumption in Mobile Phones: A Measurement Study and Implications for Network Applications." *9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC '09)* 280–93.
- Carroll, Aaron and G. Heiser. 2010. "An Analysis of Power Consumption in a Smartphone." *USENIX Annual Technical Conference* (May):21–21.
- Vallina-Rodriguez, N. 2011. "ErdOS: Achieving Energy Savings in Mobile OS." *Proceedings ACM MobiArch* 37–42.
- Loke, Seng. 2006. "Context-Aware Pervasive System: Architectures of a New Breed of Application." *IBM Journal of Research and Development* 48(5.6):617–626.
- Deligiannakis, Antonios and Yannis Kotidis. 2008. "Exploiting Spatio-Temporal Correlations For Data Processing in Sensor Networks." 45–65.
- Curran, Kevin, Eoghan Furey, T. O. M. Lunny, Jose Santos, and Derek Woods. n.d. 2011. "An Evaluation Of Indoor Location Determination Technologies." *Journal of Location Based Services* 5:61-78.
- Waqar, Wasim, Yuanzhu Chen, and Andrew Vardy. 2016. "Smartphone Positioning in Sparse Wi-Fi Environments." *Computer Communications* 73:108–17.
- Wang, Yi, Quinn A. Jacobson, Jialiu Lin, Jason Hong, and Norman Sadeh. n.d. 2009. "A Framework of Energy Efficient Mobile Sensing for Automatic User State Recognition." in *MOBISYS* 09:179 -192.
- Shi, Hongyu. 2012. "A New Weighted Centroid Localization Algorithm Based on RSSI." *2012 IEEE International Conference on Information and Automation* (June):137–41.
- WiFi Alliance. 2015. "Wi-Fi gets personal: Groundbreaking Wi-Fi Direct launches today". Press release.

- Guo, Linlin, Lei Wang, Jialin Liu, and Wei Zhou. 2017. "A Survey on Motion Detection Using WiFi Signals." *Proceedings - 12th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2016*: 202–6.
- Waqar, Wasif, Yuanzhu Chen, and Andrew Vardy. 2016. "Smartphone Positioning in Sparse Wi-Fi Environments." *Computer Communications* 73: 108–17. <http://dx.doi.org/10.1016/j.comcom.2015.09.002>.
- Baniukevic, Artur, Christian S. Jensen, and Hua Lu. 2013. "Hybrid Indoor Positioning with Wi-Fi and Bluetooth: Architecture and Performance." *Proceedings - IEEE International Conference on Mobile Data Management* 1: 207–16.
- Du, Xuan. 2018. "Map-Assisted Indoor Positioning Utilizing Ubiquitous WiFi Signals." (February). <http://repository.essex.ac.uk/21361/>.
- Wang, Yapeng et al. 2013. "Bluetooth Positioning Using RSSI and Triangulation Methods." *2013 IEEE 10th Consumer Communications and Networking Conference, CCNC 2013*: 837–42.
- Sugino, Kyohei ; Niwa, Yusuke ; Shiramatsu, Shun ; Ozono, Tadachika ; Shintani Toramatsu. 2015. "Developing a Human Motion Detector Using Bluetooth Beacons and Its Applications." *Information Engineering Express* 1(4): PP.95-105.
- Sugino, Kyohei et al. 2016. "A Bluetooth-Based Device-Free Motion Detector for a Remote Elder Care Support System." *Proceedings - 2015 IIAI 4th International Congress on Advanced Applied Informatics, IIAI-AAI 2015*: 91–96.
- Luo, Yan, Yuanzhu Peter Chen, and Orland Hoeber. 2011. "Wi-Fi-Based Indoor Positioning Using Human-Centric Collaborative Feedback." *IEEE International Conference on Communications*.
- Sebastião, Raquel, and João Gama. 2007. "Change Detection in Learning Histograms from Data Streams." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4874 LNAI(December): 112–23.
- Sri Indrawanti, Annisaa ; Wibisono, Waskitho. 2015. "A CHANGE DETECTION AND RESOURCE - AWARE DATA SENSING APPROACHES FOR IMPROVING THE REPORTING PROTOCOL MECHANISM FOR MOBILE USER." 2: 92–99.
- Ho, Shen Shyang, and Harry Wechsler. 2010. "A Martingale Framework for Detecting Changes in Data Streams by Testing Exchangeability." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(12): 2113–27.

- Mozafari, Niloofar, Sattar Hashemi, and Ali Hamzeh. 2011. "A Precise Statistical Approach for Concept Change Detection in Unlabeled Data Streams." *Computers and Mathematics with Applications* 62(4): 1655–69. <http://dx.doi.org/10.1016/j.camwa.2011.06.001>.
- Dasu, T, S Krishnan, S Venkatasubramanian, and K Yi. 2006. "An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional Data Streams." *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*: 1–24.
- Kifer, Daniel, Shai Ben-david, and Johannes Gehrke. 2004. "Detecting Change in Data Streams." : 180–91.
- Aiordachioaie, Dorel. 2014. "On Quick-Change Detection Based on Process Adaptive Modelling and Identification." *2014 International Conference on Development and Application Systems, DAS 2014 - Conference Proceedings*: 25–28.
- Kuncheva, Ludmila I. 2013. "Change Detection in Streaming Multivariate Data Using Likelihood Detectors." *IEEE Transactions on Knowledge and Data Engineering* 25(5): 1175–80.
- Duc-tan, Tran, Nguyen Dinh-chinh, Tran Duc-nghia, and Ta Duc-tuyen. 2015. "Development of a Rainfall-Triggered Landslide System Using Wireless Accelerometer Network." 7(September): 14–24.
- Sajid, Hasan and Atta Al. 2018. "Indoor Navigation to Estimate Energy Consumption in Android Platform." 3(1):32–36.
- Alvarez-Alvarez, Alberto, Jose M. Alonso, and Gracian Trivino. 2013. "Human Activity Recognition in Indoor Environments by Means of Fusing Information Extracted from Intensity of WiFi Signal and Accelerations." *Information Sciences* 233(June):162–82.

## BIODATA PENULIS



**Ade Hendi**, dilahirkan di kota Tasikmalaya, Jawa Barat pada tanggal 19 Oktober 1978. Penulis adalah anak ke tujuh dari tujuh bersaudara dan dibesarkan di kota Tasikmalaya, Jawa Barat. Penulis menempuh pendidikan SMP Negeri 4 Tasikmalaya (1992-1994), SMA Negeri 3 Cimahi (1994-1997), STMIK Mardira Indonesia Bandung (1997-2000) jurusan Komputer Akuntansi, Institut Sains Teknologi Palapa malang (2003-2005) jurusan Informatika. Pada tahun 2018, penulis diterima di strata dua pasca Jurusan Teknik Informatika Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya angkatan 2018 yang terdaftar dengan NRP. 05111850010032. Di Jurusan Teknik Informatika ini, penulis mengambil bidang minat *Mobile Computing*. Selama menempuh kuliah, penulis juga aktif sebagai IT Konsultan di Badan Pendapatan, Pengelolaan Keuangan dan Aset Daerah Kab. Gresik (2012-sekarang). Pengalaman kerja dibidang IT tahun 2004-2012 di HM Sampoerna Surabaya sebagai System Analyst. Penulis dapat dihubungi melalui alamat *e-mail* di [hendi.ade@gmail.com](mailto:hendi.ade@gmail.com).