



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN

3.1. Studi Literatur

Penelitian diawali dengan studi literatur untuk mempelajari bagaimana cara membuat aplikasi Android, protokol dan perintah yang digunakan ZigBee dalam berkomunikasi, menghubungkan konektivitas *Bluetooth* pada Android dengan perangkat ZigBee di dalam gedung, metode trilaterasi, serta penerapan semua itu dalam pembuatan mapping gedung.

3.2. Peralatan yang digunakan

Peralatan yang digunakan dalam penelitian ini meliputi perangkat keras (*hardware*) dan perangkat lunak (*software*) sebagai berikut,

a. Laptop Toshiba Satellite M300 dengan spesifikasi:

- Processor Intel® Core™ Duo CPU T850 @2.16GHz 2.17 GHz
- RAM 4096MB
- Harddisk 300GB
- Sistem Operasi Windows 7
- Terminal v1.9b
- Eclipse Indigo
- Android SDK

b. Sony Xperia S, dengan spesifikasi:

- Sistem Operasi Android 2.3. Gingerbread
- Processor Dual-Core 1.5GHz Scorpion CPU, Andreno 220 GPU, dan Qualcomm MSM8260 Snapdragon
- Memori RAM 1 GB, 32 GB kapasitas penyimpanan internal
- Resolusi layar 720 x 1280 pixel dengan ukuran 4.3 inci dan 16M-color TFT capacitive touchscreen
- Dimensi 128 x 64 x 10.6 mm
- Bluetooth v2.1 dengan A2DP dan EDR

c. Konverter *Bluetooth-to-ZigBee* yang terdiri dari:

- Bluetooth HC-06
- ZigBee ETRX2
- Kabel *jumper*
- *Breadboard*
- Input *power supply*

d. ZigBee terdiri dari 2 tipe:

- ZigBee dengan hardware ETRX2, Software ETRX3 (terpasang pada konverter)
- ZigBee dengan hardware ETRX3, Software ETRX3 (digunakan sebagai *fixed node* yang terpasang di dalam ruangan/ gedung)

e. FTDI

f. Meteran

g. Adaptor 5V untuk *power supply fixed node* ZigBee

h. Baterai untuk *power supply* konverter *Bluetooth-to-ZigBee*

3.3. Tahap-tahap Penelitian

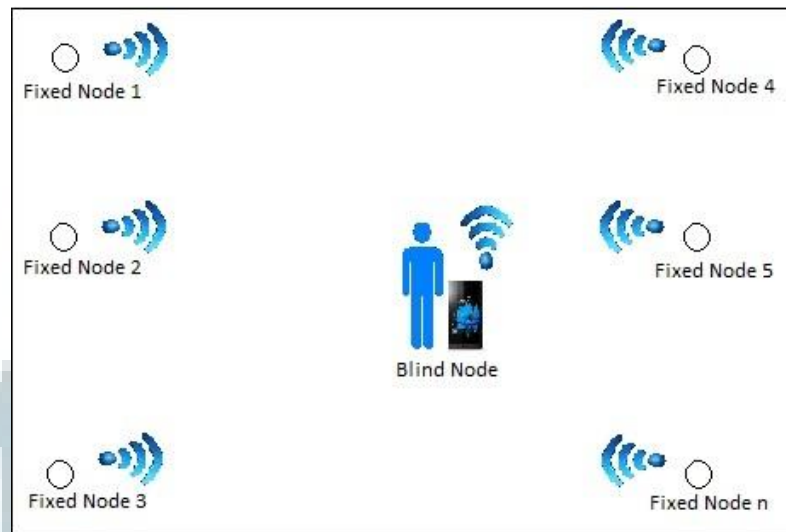
Penelitian ini dilakukan dengan tahapan mulai dari perancangan, implementasi dan pengujian.

3.3.1. Perancangan Aplikasi

Langkah-langkah dalam perancangan aplikasi sesuai urutan sebagai berikut ini,

3.3.1.1 Diagram Umum

Diagram *Localization based on ZigBee RSSI* yang digunakan pada penelitian (secara umum) terlihat pada Gambar 3.1 di bawah ini.

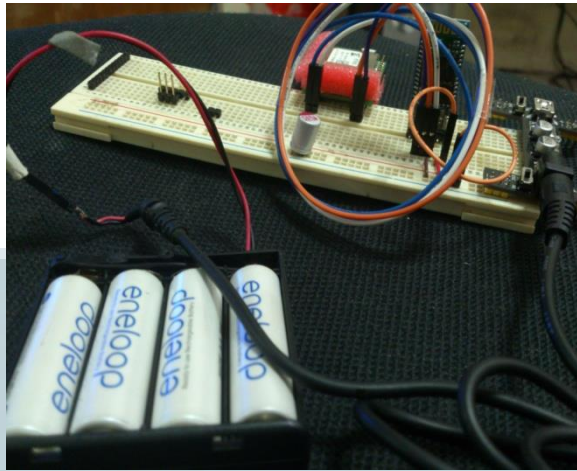


Gambar 3.1 Gambaran umum *Localization based on ZigBee RSSI* (adaptasi dari (Boon-Giin Lee, Young-Sook Lee, & Wan-Young Chung, 2008))

Sejumlah *nodes* ZigBee terpasang di dalam gedung sebagai *fixed node* (minimal 3 *nodes* untuk dapat melakukan metode trilaterasi) dan 1 *node* ZigBee sebagai *blind node*. Seluruh *fixed nodes* dan *blind node* berada dalam 1 jaringan PAN ZigBee. *Blind node* terhubung dengan perangkat Android melalui konverter *Bluetooth to ZigBee*. Metode trilaterasi digunakan untuk menghitung koordinat dari *blind node* di ruangan. Seluruh perhitungan dilakukan di perangkat Android dengan menggunakan informasi yang didapat dari *fixed node* di ruangan melalui *blind node*. Informasi yang digunakan antara lain EUI (*Extended Unique Identifier*) dan nilai RSSI dari masing-masing *fixed node* ZigBee.

3.3.1.2 Pengaturan Konektivitas antara *Bluetooth* Android dengan perangkat ZigBee

Langkah awal yang dilakukan adalah memastikan apabila perangkat Android dapat berkomunikasi dengan perangkat ZigBee di dalam gedung melalui konektivitas *Bluetooth*. Antara *Bluetooth* dan ZigBee membutuhkan perangkat konverter *Bluetooth-to-ZigBee* untuk dapat berkomunikasi, seperti yang ditampilkan pada Gambar 3.2.



Gambar 3.2 Konverter *Bluetooth-to-ZigBee*

Perangkat Android terhubung dengan konverter melalui koneksi *Bluetooth* dan terkoneksi dengan perangkat ZigBee di dalam gedung melalui konverter tersebut. Perangkat *Bluetooth* yang digunakan pada konverter adalah HC-06. Pairing dilakukan terlebih dahulu antara *Bluetooth* milik android dengan HC-06. Dikarenakan konektivitas *Bluetooth* pada Android hanya mampu untuk transfer data (*default*), maka harus dibuat koding tersendiri sehingga Android dan konverter dapat terhubung secara serial.

Aplikasi yang dibuat mengharuskan dilakukannya *pairing* manual terlebih dahulu antara *Bluetooth* milik Android dengan HC-06 pada konverter. Konektivitas *Bluetooth* antara Android dan konverter akan terhubung ketika aplikasi dijalankan. User pada Android mampu mengirimkan perintah-perintah ZigBee melalui Android ke konverter. Hal ini dikarenakan konektivitas *Bluetooth* hanya akan melewatkan perintah yang diterima pada konverter tanpa mengubah isi perintahnya dan meneruskannya ke perangkat ZigBee Telegesis ETRX-3 yang terpasang pada konverter. Perangkat Telegesis ETRX-3 ZigBee pada konverter inilah yang berkomunikasi dengan perangkat ZigBee yang terpasang di dalam gedung.

3.3.1.3 Pengaturan Jaringan ZigBee

Langkah berikut yang dilakukan adalah pengaturan jaringan ZigBee di dalam gedung yang akan digunakan dalam proses *mapping*. Proses *mapping* yang menggunakan metode trilaterasi ini mengharuskan setidaknya ada 3 *node* ZigBee yang sudah diketahui lokasinya di dalam gedung. Ketiga *node* ZigBee tersebut harus berada pada 1 jaringan (PAN). Tahap-tahap yang dilakukan di sini antara lain:

1. Menyiapkan 6 *node* ZigBee yang akan digunakan, dalam hal ini digunakan ZigBee Telegesis ETRX-3.
2. Membuat salah satu *node* ZigBee menjadi koordinator (COO) dan membentuk jaringan (PAN) ZigBee yang akan digunakan nantinya.
3. Menggabungkan *node* ZigBee lainnya ke dalam jaringan (PAN) yang sudah dibuat menggunakan perintah AT+JPAN, sehingga semua *node* berada dalam satu jaringan sekarang. AT+JPAN: membuat *node* ZigBee untuk bergabung dengan PAN (jaringan) tertentu.
4. Mencatat EUI (*Extended Unique Identifier*) dari masing-masing *node* ZigBee yang digunakan untuk penggunaan ke depannya. EUI sama seperti *Mac Address* yang ada pada komputer.
5. Mengatur Bit C dari S10 masing-masing *node* ZigBee sehingga nilai RSSI (*Received Signal Strength Indication*) dan LQI (*Link Quality Index*) akan ditampilkan ketika perintah AT+SN dipanggil. AT+SN: memunculkan list dari semua *node* ZigBee yang berada di dalam satu jaringan (PAN) dan berada dalam jangkauan. Data yang akan diambil dari AT+SN adalah EUI dan RSSI dari masing-masing *node*.
6. Mengatur penempatan dari semua *node* ZigBee di dalam gedung dan dicatat lokasinya masing-masing.

Pengaturan-pengaturan di atas dilakukan melalui Terminal pada komputer yang terhubung dengan *node* ZigBee yang akan di-*setting* yang melibatkan beberapa perintah-perintah (command) ZigBee. Daftar dari perintah-perintah yang digunakan adalah:

1. AT+JPAN: menggabungkan perangkat ZigBee ke dalam jaringan (PAN) tertentu.
2. AT+SN: memunculkan daftar dari semua *node* ZigBee yang berada di dalam satu jaringan (PAN) dan berada dalam jangkauan. Perintah ini sekaligus menampilkan nilai RSSI apabila Bit C dari S10 ZigBee diaktifkan.
3. AT+DASSL: membuat perangkat ZigBee keluar dari jaringan (PAN).
4. AT+PANSCAN: meng-*scan* dan memunculkan daftar dari semua jaringan (PAN) yang berada di dalam jangkauan.
5. AT+EN: membuat jaringan (PAN) baru.

Berikut adalah foto dari perangkat ZigBee Telegesis ETRX-3 yang digunakan.



Gambar 3.3 Telegesis ETRX-3 dengan port USB

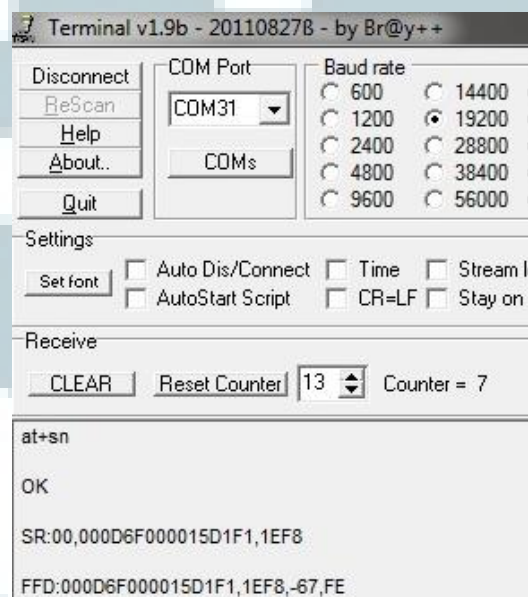
3.3.1.4 Pembuatan aplikasi untuk mengambil nilai RSSI dari ZigBee

Setelah jaringan ZigBee di dalam gedung sudah siap dan Android dapat berkomunikasi dengan ZigBee melalui konverter *Bluetooth-to-ZigBee*, langkah berikutnya adalah pembuatan aplikasi untuk mengolah data yang diterima Android dari komunikasi tersebut. Sebelum ZigBee yang terpasang dalam gedung mengirimkan data ke ZigBee yang berada pada konverter, perintah AT+SN harus

dikirimkan terlebih dahulu dari Android ke ZigBee melalui konverter. Ketika perintah AT+SN diisukan, ZigBee yang terpasang pada konverter *Bluetooth-to-ZigBee* akan memunculkan daftar dari seluruh *node* ZigBee yang berada di dalam jaringannya dan berada dalam jangkauan. Daftar tersebut terdiri dari:

1. Tipe dari *node*
2. EUI (*Extended Unique Identifier*)
3. Node ID
4. RSSI (dalam dBm)
5. LQI (dalam heksadesimal)

Berikut adalah contoh dari data yang diterima ketika AT+SN diisukan:



Gambar 3.4 Data yang diterima oleh terminal ketika AT+SN diisukan

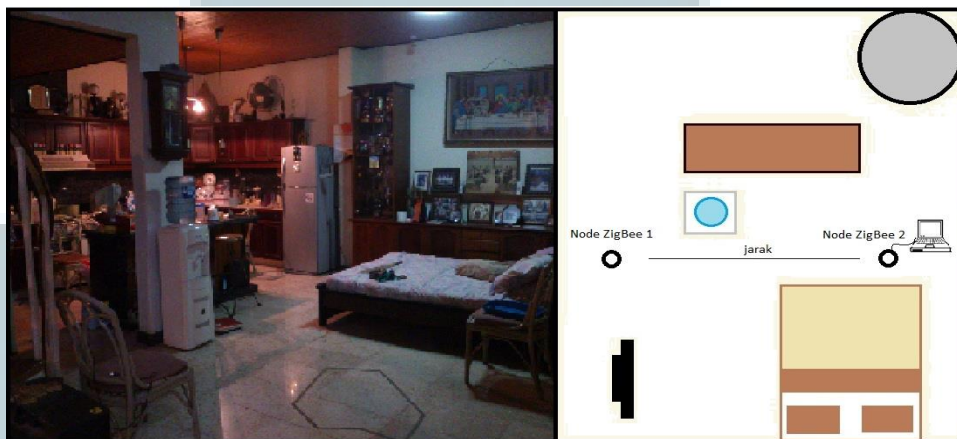
FFD merupakan tipe dari perangkat ZigBee masing-masing, 000D6F000015D1F1 merupakan EUI, -67 merupakan nilai RSSI, dan FE merupakan nilai dari LQI. Aplikasi yang dibuat dengan memanfaatkan EUI dan RSSI dari data yang didapat. EUI yang didapat dicocokkan dengan data dari EUI masing-masing *node* ZigBee yang terpasang di dalam gedung dan posisinya masing-masing, Kemudian setelah cocok, dicek nilai RSSI dari *node* tersebut yang ke depannya akan dipakai untuk perhitungan trilaterasi. Dibutuhkan setidaknya 3 EUI dan RSSI-nya masing-

masing untuk menghitung lokasi. Keseluruhan aplikasi dibuat menggunakan Eclipse IDE dan kemudian aplikasi tersebut dijalankan di perangkat Android.

3.3.1.5 Pembuatan rumus konversi RSSI ZigBee ke jarak

Pembuatan rumus konversi RSSI (kuat sinyal) ZigBee ke dalam jarak dilakukan menggunakan aplikasi yang telah dibuat dan meneliti berapa kuat sinyal antara 2 *node* ZigBee pada jarak tertentu. Penelitian dilakukan untuk mengambil *sample* sebanyak-banyaknya dari masing-masing jarak.

Pada penelitian yang dilakukan, 2 *nodes* ZigBee diletakkan terpisah pada jarak tertentu. 1 *node* terpasang pada laptop dan *node* lainnya diletakkan pada jarak tertentu dari *node* pertama. Antara kedua *node* tersebut tidak ada penghalang apapun. *Input* pada pengujian ini adalah jarak, berkisar antara 1-8 meter. Hasil *output* yang diamati adalah nilai RSSI (kuat sinyal) antara kedua *nodes* ZigBee tersebut. Data yang sudah didapat dimasukkan ke dalam aplikasi MATLAB untuk dicari grafik hubungan antara jarak dan RSSI dan dicari formulanya. Formula tersebut akan digunakan di aplikasi untuk mengkonversi nilai RSSI antara *blind node* dengan *fixed node* yang terpasang di gedung ke dalam jarak.



Gambar 3.5 Ruang tempat pengujian

3.3.1.6 Pembuatan aplikasi untuk menentukan koordinat lokasi

Setelah formula untuk mengkonversi RSSI ZigBee ke dalam jarak sudah didapat, kemudian dibuat aplikasi untuk menentukan koordinat lokasi dari sejumlah jarak dari semua *fixed node* ZigBee menggunakan formula trilaterasi

seperti yang sudah dijelaskan di Bab 2. Aplikasi dibuat pada perangkat Android dan melanjutkan aplikasi yang sudah dibuat sebelumnya (aplikasi untuk mengambil nilai RSSI dari ZigBee). Dalam aplikasi ini, nilai RSSI terhadap masing-masing *fixed nodes* ZigBee diambil dan kemudian dikonversikan ke dalam jarak (meter) menggunakan formula yang ditemukan. Dari formula tersebut, maka diperoleh sejumlah jarak dari lokasi ke semua *fixed nodes* ZigBee yang terpasang yang sudah diketahui posisinya masing-masing. Dalam penelitian ini, dilakukan pengujian dengan menggunakan 6 buah *node* ZigBee.

3.3.1.7 Pembuatan User Interface untuk mapping posisi lokasi

User interface dibuat pada perangkat Android Xperia S. Dengan menggunakan koordinat yang didapat dan dengan menyesuaikan resolusi layar pada perangkat Android, *user interface* dibuat menggunakan denah ruangan/gedung sebagai *background* dan sebuah pointer yang berpindah-pindah tergantung dari koordinat yang didapat. Posisi pointer selalu di-*update* setiap kali koordinat didapat dari perhitungan dan kemudian perangkat Android meng-*request* nilai RSSI yang baru dari jaringan ZigBee (dengan perintah AT+SN) untuk perhitungan berikutnya. Proses di atas dilakukan secara terus menerus sampai koneksi *Bluetooth* Android dengan *Bluetooth* konverter dihentikan. Dengan sistem seperti itu, dapat dihasilkan *mapping* yang *updated* setiap saat.

3.4. Implementasi Aplikasi

Aplikasi diimplementasikan di ruangan/ gedung yang memiliki jaringan ZigBee. Sebelum implementasi, setidaknya harus ada 3 *node* ZigBee yang terpasang dalam 1 jaringan dan masing-masing *node* diketahui posisinya.

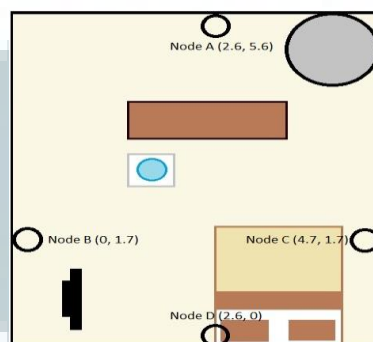
3.5. Pengujian

Untuk mengetahui aplikasi berfungsi dengan baik, dilakukan pengujian tingkat keakurasian perhitungan koordinat lokasi dibandingkan dengan posisi lokasi sebenarnya.

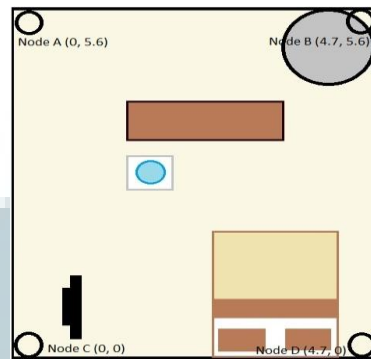
Pengujian dilakukan dengan menggunakan aplikasi Android yang sudah selesai dan diuji untuk melihat keakurasian dari perhitungan koordinat berdasarkan RSSI ZigBee. Di dalam pengujian, aplikasi dibuat agar tidak meminta nilai RSSI secara terus-menerus, melainkan harus di-*triggered* oleh *button* terlebih dahulu. Hal ini dilakukan agar setiap nilai RSSI, jarak, dan koordinat yang dihasilkan dapat dimonitor dan dicatat sehingga dapat dibandingkan dengan kondisi sebenarnya. Tahapan pengujian yang dilakukan adalah sebagai berikut,

a. Pengujian tingkat keakurasian perhitungan koordinat lokasi

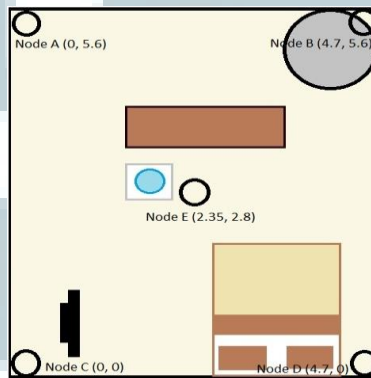
Pengujian dilakukan beberapa kali dengan menggunakan 6 buah *node* ZigBee. Pengujian dilakukan untuk menghitung tingkat keakurasian perhitungan koordinat lokasi menggunakan metode trilaterasi dari 3 jarak yang diperoleh dari 3 *fixed nodes* atau lebih yang terpasang di dalam ruangan/ gedung. *Input* pada pengujian ini adalah 27 titik koordinat dan koordinat semua *fixed nodes* ZigBee (lihat pada lampiran nomor 4). *Output* pada pengujian ini adalah koordinat *blind node* hasil perhitungan. Berikut adalah denah sederhana dari ruang tempat pengujian yang berukuran 4.7m x 5.6m beserta koordinat masing-masing *fixed nodes* ZigBee di setiap pengujian.



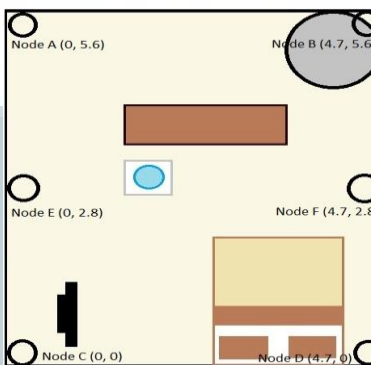
Gambar 3.6 Denah ruang pengujian 1



Gambar 3.7 Denah ruang pengujian 2

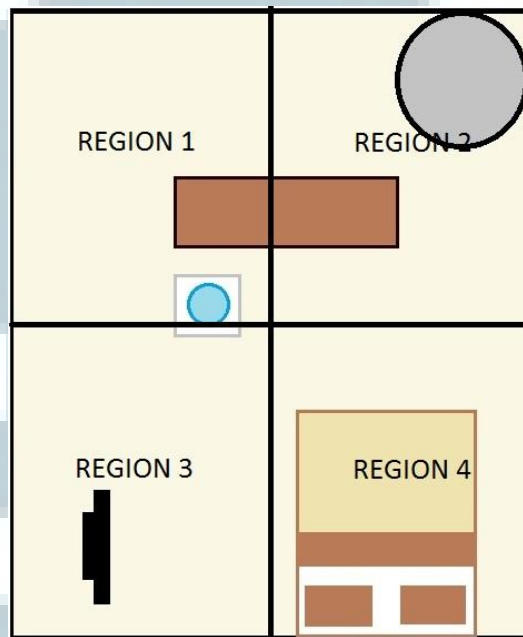


Gambar 3.8 Denah ruang pengujian 3



Gambar 3.9 Denah ruang pengujian 4

Ruang dibagi menjadi 4 region, yaitu region 1,2,3, dan 4. Sebuah perhitungan dianggap akurat apabila koordinat hasil perhitungan dan koordinat asli berada di dalam 1 region.



Gambar 3.10 Ruang dibagi menjadi 4 region

b. Pengujian *user interface* aplikasi

Pengujian dilakukan untuk menguji apabila *user interface* aplikasi (*mapping*) menunjukkan lokasi yang sesuai dengan koordinat inputan atau tidak. Aplikasi diujikan pada perangkat Android Xperia S. Inputan yang dimasukkan adalah koordinat lokasi pada Tabel 3.1. *Output* pada pengujian ini adalah *screenshot* dari *user interface*.

Tabel 3.1 Input koordinat pada pengujian *user interface*

No.	x (meter)	y (meter)
1	0	5.6
2	4	5.6
3	0	0
4	4.2	0
5	2.5	3

c. Pengujian waktu *update* posisi

Pengujian dilakukan untuk mengetahui berapa lama waktu yang diperlukan oleh aplikasi setiap kali melakukan *update* posisi lokasi dan menampilkannya di dalam map. *Input* pada pengujian ini adalah proses *update* posisi lokasi dalam aplikasi. *Output* pada pengujian ini adalah jangka waktu yang diperlukan setiap kali posisi lokasi di-*update* di dalam aplikasi *mapping*.

