

BAB III

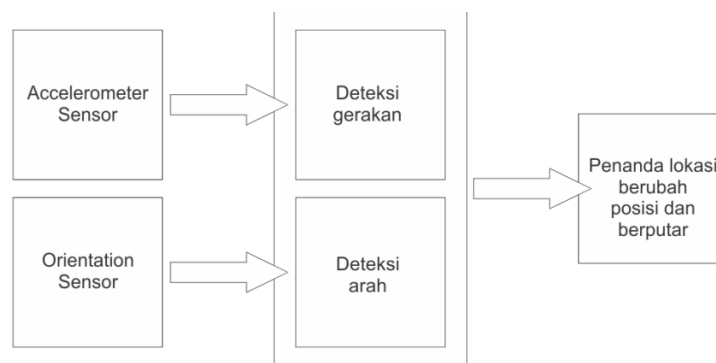
METODE PENELITIAN

3.1 Metode Penelitian

Metodologi penelitian yang digunakan dalam Tugas Akhir ini adalah dengan mengakses Sensor *Accelerometer* dan Sensor Orientasi pada *smartphone* yang digunakan, kemudian mengolah data dari kedua sensor tersebut. Data dari Sensor *Accelerometer* akan digunakan sebagai deteksi gerakan, dengan cara mengklasifikasikan data tersebut dalam tiga golongan, yaitu diam, berjalan dengan langkah normal, dan berlari. Golongan-golongan ini nantinya akan menentukan seberapa jauh pergerakan *pixel* penanda lokasi pada *minimap*. Data dari Sensor Orientasi akan digunakan sebagai deteksi arah, dengan cara membaca nilai *azimuth* yang terbaca dan kemudian mengklasifikasikan data tersebut ke dalam 8 arah mata angin. Arah mata angin yang terpilih akan menentukan arah pergeseran penanda lokasi pada *minimap*.

3.2 Blok Diagram Sistem

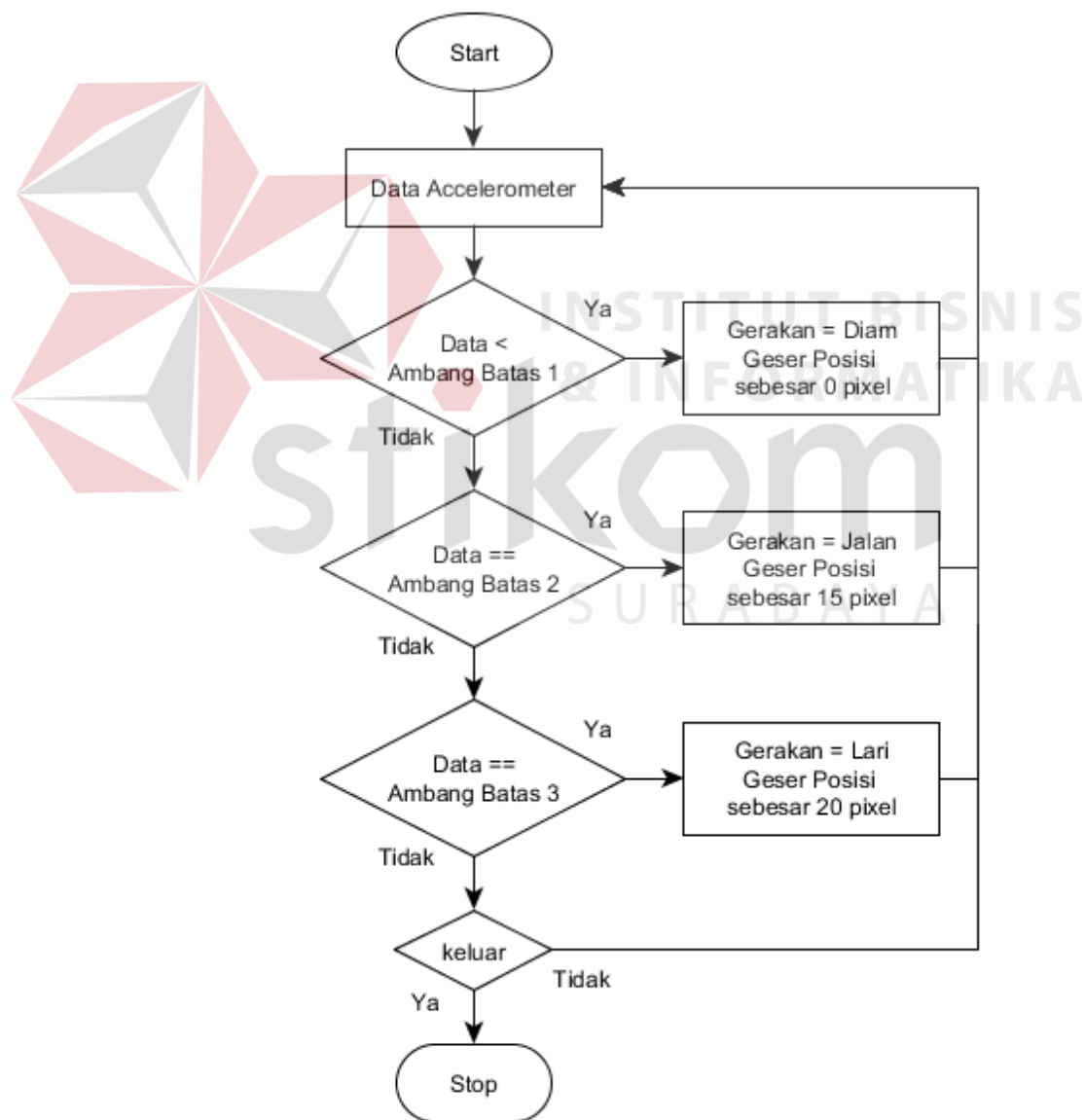
Adapun perancangan blok diagram sistem ditunjukkan pada Gambar 3.1



Gambar 3.1 Blok Diagram Sistem

3.2.1 Deteksi Gerakan

Sensor *Accelerometer* akan mendeteksi apakah penguji sedang diam, berjalan dengan langkah normal, atau berlari. Pendeteksian ini dilakukan dengan menggolongkan nilai puncak Y yang diterima dari sensor *accelerometer*, yaitu apabila nilai yang terbaca kurang dari ambang batas 1, maka penguji diasumsikan diam.



Gambar 3.2 Proses Deteksi Gerakan

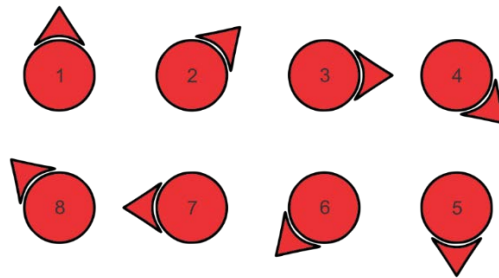
Apabila nilai yang terbaca sama dengan ambang batas 2, maka penguji diasumsikan berjalan dengan langkah normal dan penanda lokasi akan digeser sebesar 15px, dan apabila nilai yang terbaca sama dengan ambang batas 3, maka penguji diasumsikan berlari dan penanda lokasi digeser sebesar 20px. Digunakannya kondisi sama dengan (=) pada proses deteksi gerakan ini dimaksudkan agar mengurangi terjadinya kesalahan jumlah langkah yang terdeteksi. Kesalahan yang dimaksud adalah deteksi ganda dalam satu langkah. Apabila menggunakan kondisi lebih dari (>) ambang batas, maka untuk setiap nilai yang ada diatas ambang batas akan terhitung sebagai satu langkah.

3.2.2 Deteksi Arah

Orientation sensor akan mendeteksi arah hadap penguji dengan menggunakan nilai *azimuth*. Nilai *azimuth* yang didapatkan dari sensor akan digolongkan menjadi 8 bagian dalam arah mata angin. Pembagian 8 arah mata angin adalah seperti berikut:

1. Utara (U) : $327.5^{\circ} - 360^{\circ}$ dan $0^{\circ} - 22.5^{\circ}$
2. Timur laut (TL) : $22.5^{\circ} - 67.5^{\circ}$
3. Timur (T) : $67.5^{\circ} - 112.5^{\circ}$
4. Tenggara (TG) : $112.5^{\circ} - 157.5^{\circ}$
5. Selatan (S) : $157.5^{\circ} - 202.5^{\circ}$
6. Barat daya (BD) : $202.5^{\circ} - 247.5^{\circ}$
7. Barat (B) : $247.5^{\circ} - 292.5^{\circ}$
8. Barat laut (BL) : $292.5^{\circ} - 327.5^{\circ}$

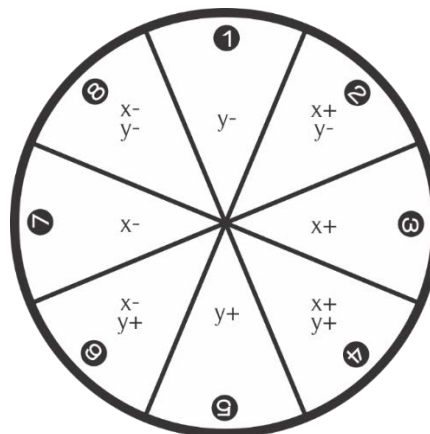
Nilai *azimuth* yang diperoleh dari *orientation sensor* akan dikonversikan menjadi rotasi penanda lokasi sesuai dengan bagian-bagian yang disebutkan diatas. Rotasi penanda lokasi akan tampak seperti Gambar 3.3 berikut.



Gambar 3.3 Rotasi Penanda Lokasi

3.2.3 Perubahan Posisi Dan Arah Penanda Lokasi

Tampilan dalam aplikasi *smartphone* menggunakan sistem koordinat. Jadi untuk pergerakan suatu objek dalam tampilan menggunakan koordinat x dan y dengan titik ujung kiri atas layar sebagai pusat koordinat $(0,0)$. Berdasarkan sistem koordinat tersebut, maka hasil dari proses deteksi arah dan deteksi gerakan akan diterapkan menjadi perubahan posisi dan arah penanda lokasi seperti pada Gambar 3.4.



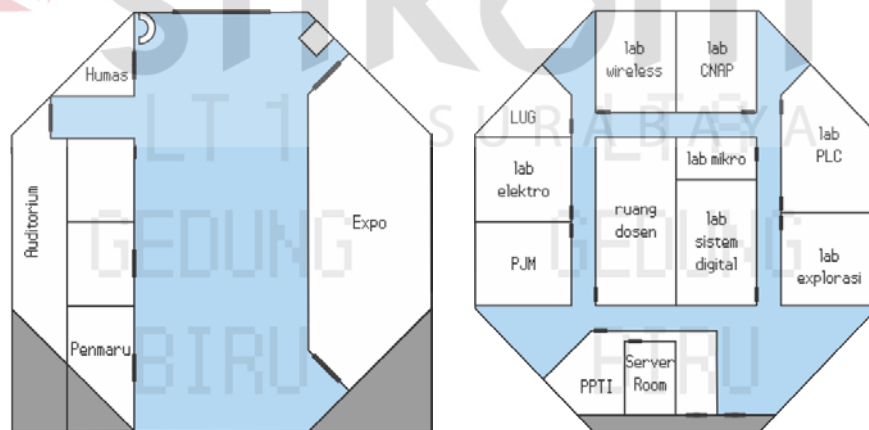
Gambar 3.4 Perubahan Posisi Berdasarkan 8 Arah Mata Angin

3.3 Perancangan Aplikasi

Perancangan aplikasi “*Personal Location Detection*” ini dibagi menjadi 4 bagian yaitu sebagai berikut:

3.3.1 Perancangan Peta

Pada aplikasi ini, peta yang akan digunakan adalah peta Lantai 1 Gedung Biru dan Lantai 8 Gedung Biru Institut Bisnis dan Informatika Stikom Surabaya. Peta dirancang sepresisi mungkin menggunakan perangkat lunak (*software*) desain grafis 2 dimensi, sebagai contoh, CorelDraw, Adobe Illustrator, dan Inkscape. Peta digambar dengan dimensi 500x500 *pixel* dengan skala 1:5, satu *pixel* pada peta setara dengan 5cm pada lapangan. Jadi, dengan asumsi tiap langkah 50cm, maka dari ujung ke ujung dapat ditempuh dalam 50 langkah. Berikut rancangan peta yang akan digunakan dalam aplikasi.

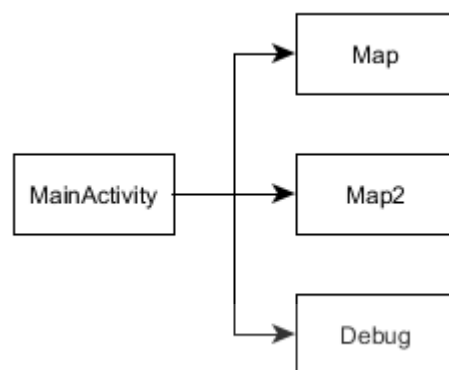


Gambar 3.5 Peta Lantai 1 Dan Lantai 8 Gedung Biru

Bagian-bagian dalam peta dan besar dimensinya didapatkan dari pengujian lapangan pada Lantai 1 Gedung Biru dan Lantai 8 Gedung Biru.

3.3.2 Perancangan Alur Aplikasi

Aplikasi perlu dirancang alur jalannya agar mudah dipahami oleh pembaca. Alur aplikasi dapat dilihat pada Gambar 3.6 berikut:



Gambar 3.6 Alur Aplikasi

Alur aplikasi diatas menunjukkan alur *Activity* yang terdapat dalam aplikasi. *MainActivity* merupakan *activity* untuk pemilihan peta yang akan dimuat, yaitu Map (Lantai 1 Gedung Biru), Map2 (Lantai 8 Gedung Biru). Pada *MainActivity* juga terdapat pilihan untuk masuk ke mode *Debug*, yaitu mode untuk menguji kepekaan pendeteksian gerakan dengan ambang batas yang digunakan, apakah sudah berjalan sesuai dengan harapan atau tidak.

3.3.3 Perancangan Program Aplikasi

Perancangan program aplikasi ditujukan agar aplikasi yang dibuat nantinya akan memiliki baris pengkodean yang terstruktur. Perancangan struktur program dilakukan dengan mengubah file berekstensi java. Beberapa fungsi kunci yang digunakan dalam pemrograman ini akan diuraikan sebagai berikut:

a) Pengaksesan Sensor

Sebelum dapat digunakan, sensor-sensor yang akan digunakan harus didaftarkan (*register*) terlebih dahulu dengan menggunakan fungsi *Sensor Manager*.

```
Sensor ACL, ORI;
SensorManager SM;

:

//register sensor
SM=(SensorManager) getSystemService (SENSOR_SERVICE);
ACL= SM.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
ORI= SM.getDefaultSensor(Sensor.TYPE_ORIENTATION);
SM.registerListener(Map.this, ORI, SensorManager.SENSOR_DELAY_GAME,
    SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
SM.registerListener(Map.this, ACL, SensorManager.SENSOR_DELAY_NORMAL,
    SensorManager.SENSOR_STATUS_ACCURACY_HIGH);
```

Gambar 3.7 Register Sensor Accelerometer Dan Sensor Orientasi

Pertama-tama sensor harus diberikan sebuah variabel yang akan menampung data, dalam hal ini adalah variabel ACL dan ORI. Kemudian dengan menggunakan *Sensor Manager*, variabel ACL diatur untuk data dari sensor *accelerometer*, dan variabel ORI untuk data dari sensor orientasi. Setelah itu, sensor juga perlu untuk diatur besar *delay* tiap *event* dan tingkat akurasi.

b) Deteksi Arah

Data yang dikirim oleh sensor diterima secara serentak melalui sebuah fungsi `public void onSensorChanged (SensorEvent event)`, jadi untuk mendapatkan data hanya dari sensor orientasi, maka perlu disaring dengan menggunakan fungsi `event.sensor.getType() ==`

Sensor.TYPE_ORIENTATION. Untuk lebih jelasnya, perhatikan Gambar 3.8 berikut:

```
if (event.sensor.getType()==Sensor.TYPE_ORIENTATION) {
    orie = event.values.clone();
    angle=orie[0];
    orient.setText("Orientation:\n" + "Azimuth: " +Math.round(orie[0]));
    //deteksi arah dan konversi rotasi
    if (angle >= 337.5 || angle <22.5) {
        arrow.setRotation(0);dir='1';}
    else if(angle >= 22.5 && angle <67.5) {
        arrow.setRotation(45);dir='2';}
    else if(angle >= 67.5 && angle <112.5) {
        arrow.setRotation(90);dir='3';}
    else if(angle >= 112.5 && angle <157.5) {
        arrow.setRotation(135);dir='4';}
    else if(angle >= 157.5 && angle <202.5) {
        arrow.setRotation(180);dir='5';}
    else if(angle >= 202.5 && angle <247.5) {
        arrow.setRotation(225);dir='6';}
    else if(angle >= 247.5 && angle <292.5) {
        arrow.setRotation(270);dir='7';}
    else if(angle >= 292.5 && angle <337.5) {
        arrow.setRotation(315);dir='8';}
}
```

Gambar 3.8 Klasifikasi Nilai Dari Sensor Orientasi

Data yang digunakan adalah nilai *azimuth* yang disimpan pada variabel `orie[0]`. Fungsi `Math.round` digunakan untuk pembulatan nilai, karena nilai yang didapatkan dari sensor memiliki tipe data *float*, agar memudahkan pengolahan data, maka nilai dijadikan nilai bulat (*integer*). Penggolongan perubahan arah penunjuk lokasi disesuaikan dengan 8 arah mata angin. Jangkauan-jangkauan sudut tersebut didapatkan dari rumus berikut:

$$360^{\circ}/8 = 45^{\circ}$$

Jadi setiap jangkauan memiliki besar sudut 45 derajat. Untuk menunjukkan posisi utara, maka:

$$45^{\circ}/2 = 22,5^{\circ}$$

Jadi jangkauan sudut untuk menunjuk arah utara adalah:

$$0^{\circ} + 22,5^{\circ} = 22,5^{\circ}$$

$$360^{\circ} - 22,5^{\circ} = 337,5^{\circ}$$

Jika masuk dalam suatu jangkauan, maka penunjuk lokasi akan diputar dengan fungsi `setRotation` dan diberikan penanda `dir` sesuai dengan jangkauan.

c) Deteksi Gerakan

Untuk mendapatkan data hanya dari sensor *accelerometer*, maka fungsi yang perlu digunakan pada `public void onSensorChanged (SensorEvent event)` untuk menyaring data adalah `event.sensor.getType() == Sensor.TYPE_ORIENTATION`. Untuk lebih jelasnya, perhatikan Gambar 3.9 berikut:

```
if(event.sensor.getType() == Sensor.TYPE_ACCELEROMETER) {
    accel = event.values.clone();
    acclY=Math.round(accel[1]);
    acclZ=accel[2];
    //penentuan threshold untuk diam, langkah normal, dan lari
    if (acclY > 3 && acclZ > 11) { //diam
        if (acclY == 4 && moved==false) { //langkah normal
            if(dir=='1'){
                arrow.setY(arrow.getY() - 15); //pergerakan penanda lokasi
            }
            else if (dir=='2'){
                arrow.setX(arrow.getX() + 15);
                arrow.setY(arrow.getY() - 15);
            }
        }
    }
}
```

Gambar 3.9 Klasifikasi Nilai Dari Sensor *Accelerometer*

Data yang digunakan adalah nilai percepatan sumbu Y yang disimpan pada variabel `ac1Y` dan nilai percepatan sumbu Z yang disimpan pada variabel `ac1Z`. Fungsi `Math.round` digunakan untuk pembulatan nilai, karena nilai yang didapatkan dari sensor memiliki tipe data *float*, agar memudahkan pengolahan data, maka nilai dijadikan nilai bulat (*integer*). Pengguna diasumsikan diam apabila gerakan yang dilakukan tidak menyebabkan nilai `ac1Y` dan `ac1Z` melebihi ambang batas 1 yaitu `ac1Y` lebih dari 3 dan `ac1Z` lebih dari 11. Apabila telah melebihi, akan dicek kembali, apakah termasuk dalam gerakan langkah normal atau lari. Ambang batas 2 yang menentukan bahwa pengguna berjalan normal adalah ketika `ac1Y==4`. Sedangkan ambang batas 3 yang menentukan bahwa pengguna berlari adalah ketika `ac1Y==7`. Apabila kondisi terpenuhi, maka akan dicek berapakah nilai `dir` yang didapatkan dari proses deteksi arah. Nilai `dir` menentukan bagaimana pergeseran penanda lokasi pada *minimap*. Terdapat sedikit perbedaan pada *activity* Debug. Pada *activity* ini, terdapat variabel `calibration`.

```
calibrate.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        calibration=Math.round(accel[1]);
    }
});

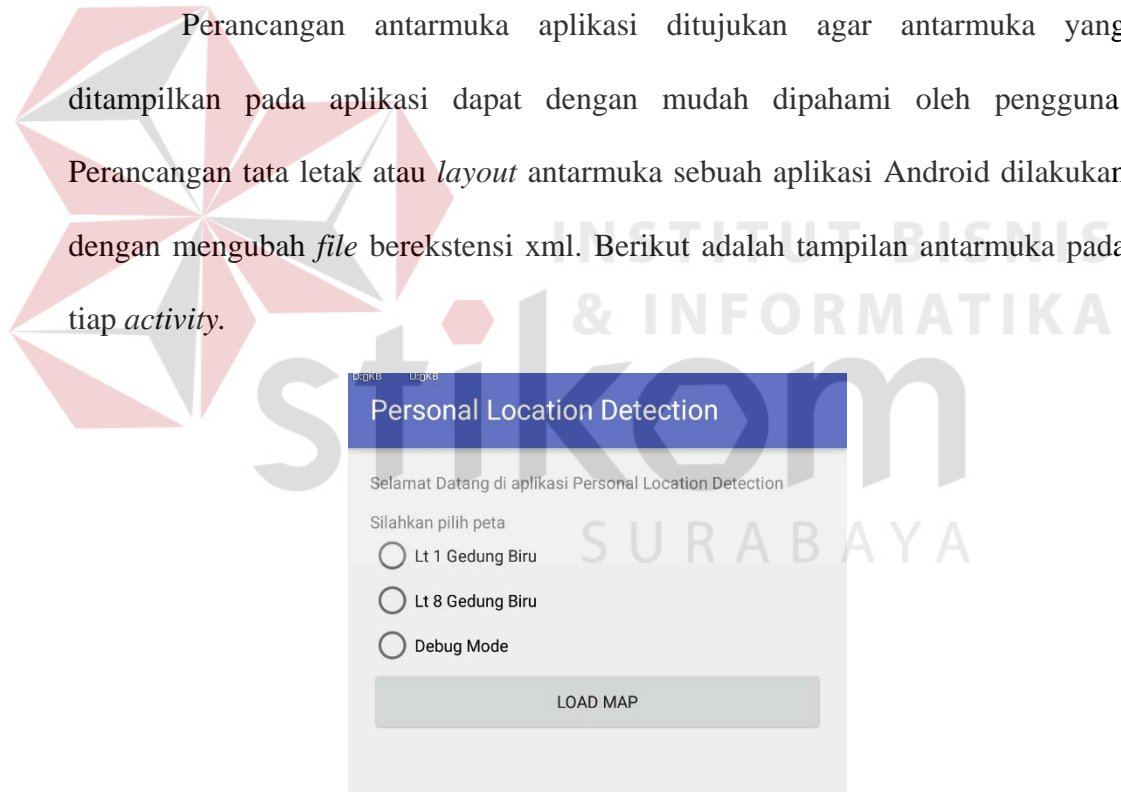
if (ac1Y > calibration && ac1Z > 11) {
    if (ac1Y == calibration+1 && moved==false) {
        totaljln++;
        Jalan.setText("Jalan:\n" + totaljln);
        moved=true;
    }
    if (ac1Y == calibration+4) {
        totallari++;
        Lari.setText("Lari:\n" + totallari);
        moved=true;
    }
}
```

Gambar 3.10 Variabel *Calibration* Pada *Activity* Debug

Variabel ini digunakan sebagai ambang batas dinamis, dengan ambang batas 1 $acly > calibration$, ambang batas 2 adalah $acly == calibration+1$ dan ambang batas 3 adalah $acly == calibration+4$. Ambang batas dinamis ini dimaksudkan untuk mengetahui dengan dasar kalibrasi manakah yang memungkinkan deteksi gerakan secara maksimal.

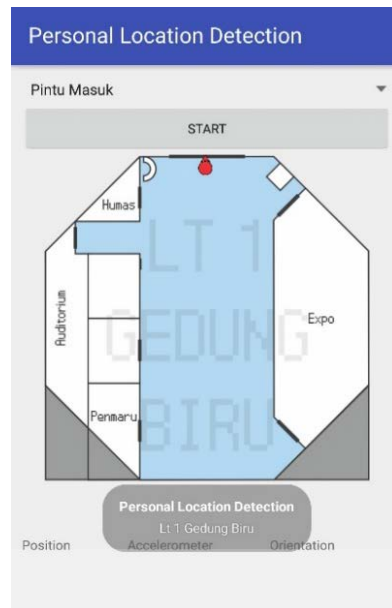
3.3.4 Perancangan Antarmuka Aplikasi

Perancangan antarmuka aplikasi ditujukan agar antarmuka yang ditampilkan pada aplikasi dapat dengan mudah dipahami oleh pengguna. Perancangan tata letak atau *layout* antarmuka sebuah aplikasi Android dilakukan dengan mengubah *file* berekstensi xml. Berikut adalah tampilan antarmuka pada tiap *activity*.



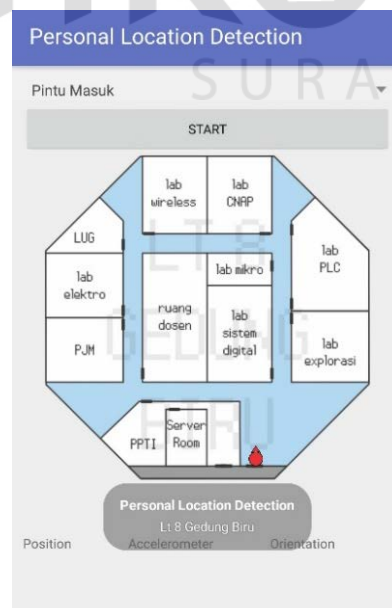
Gambar 3.11 Antarmuka *Activity MainActivity*

Antarmuka *MainActivity* menyediakan pilihan untuk berpindah ke Peta Lantai 1 Gedung Biru (*activity Map*), Peta Lantai 8 Gedung Biru (*activity Map2*), dan Debug Mode (*activity Debug*).



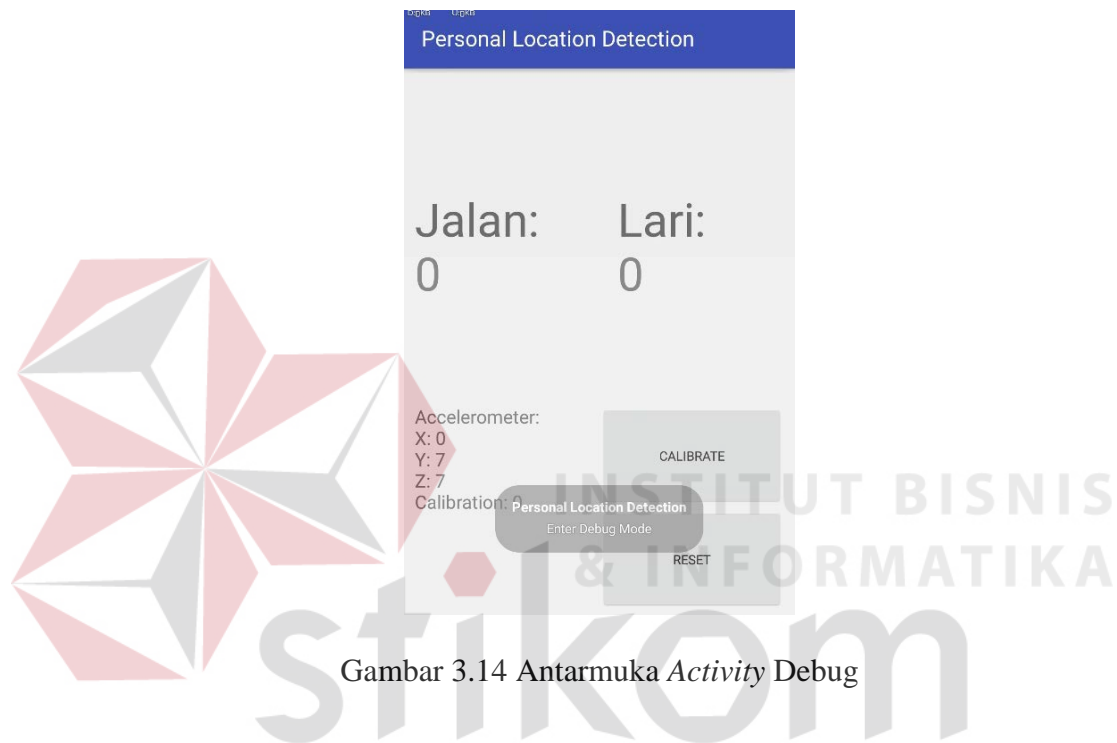
Gambar 3.12 Antarmuka Activity Map

Antarmuka Map menampilkan Peta Lantai 1 Gedung Biru, disertai pilihan lokasi mulai pengujian, koordinat penunjuk lokasi pada peta, nilai dari *accelerometer sensor* dan nilai dari *orientation sensor*.



Gambar 3.13 Antarmuka Activity Map2

Antarmuka *activity* Map2 tidak jauh berbeda dengan *activity* Map, menampilkan Peta Lantai 8 Gedung Biru, disertai pilihan lokasi mulai penguji, koordinat penunjuk lokasi pada peta, nilai dari *accelerometer sensor* dan nilai dari *orientation sensor*.



Gambar 3.14 Antarmuka *Activity* Debug

Antarmuka *activity* Debug menampilkan nilai dari *accelerometer sensor*, jumlah langkah normal dan lari yang terdeteksi, nilai kalibrasi, tombol reset dan tombol kalibrasi. Kalibrasi dalam *activity* ini akan digunakan untuk menentukan posisi awal *smartphone* ketika dipegang oleh penguji. Sehingga ambang batas yang digunakan dapat digunakan sepresisi mungkin.

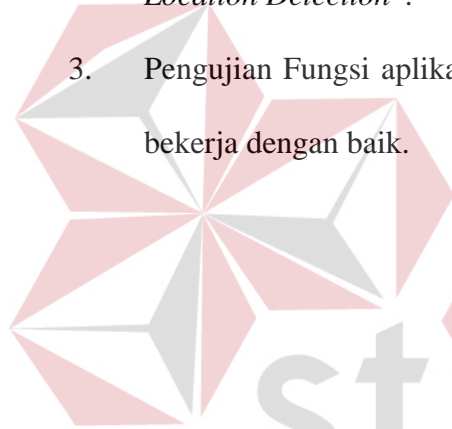
3.4 Metode Analisis

Metodologi analisis yang digunakan dalam penelitian Tugas Akhir ini adalah dengan menganalisis perbedaan kondisi akhir penanda lokasi pada aplikasi

dan kondisi penguji secara fisik, apakah telah sesuai antara keduanya atau tidak.

Kondisi akhir tersebut didasarkan pada tiga tipe pengujian, yaitu:

1. Pengujian *Accelerometer Sensor*: jumlah langkah normal dan lari yang terdeteksi oleh aplikasi setelah penguji melakukan 3 skenario pengujian, yaitu berjalan dengan langkah normal, berlari, dan campuran langkah jalan dan lari.
2. Pengujian *Orientation Sensor*: sudut arah uji pada kompas dibandingkan dengan sudut arah penanda lokasi yang terdapat pada aplikasi “*Personal Location Detection*”.
3. Pengujian Fungsi aplikasi: fungsi-fungsi yang terdapat dalam aplikasi dapat bekerja dengan baik.



INSTITUT BISNIS
& INFORMATIKA
stikom
SURABAYA