BAB 3
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

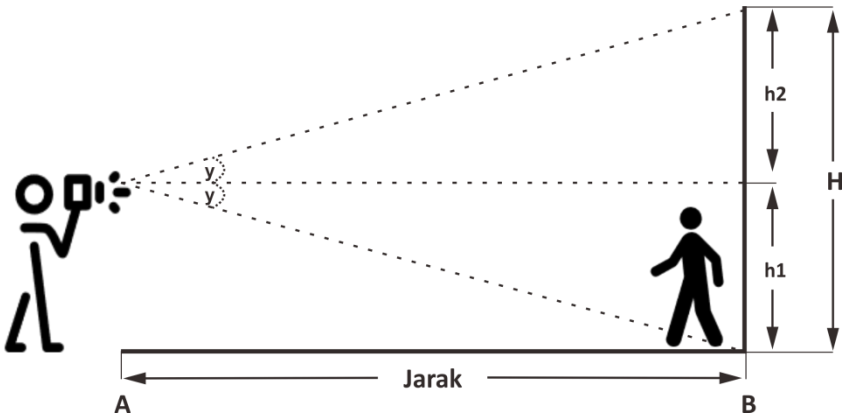
Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang terdapat pada sistem serta menentukan kebutuhan-kebutuhan dari sistem yang dibangun. Analisis tersebut meliputi analisis masalah, analisis metode, analisis kebutuhan non-fungsional, dan analisis kebutuhan fungsional.

3.1.1 Analisis Masalah

Analisis masalah sangat penting dalam suatu penelitian, karena pada tahapan ini merupakan salah satu proses mengidentifikasi masalah-masalah yang berkaitan dengan bagaimana membangun aplikasi pengukur ketinggian dan kemiringan objek menggunakan *accelerometer* berbasis *android*. Berdasarkan latar belakang masalah pada bab 1 yang telah diuraikan, yang menjadi permaslahan dalam penelitian ini yaitu penggunaan meteran tidak efektif, seperti kasus yang sedang dialami oleh perusahaan CV. PROJECTRON CORPORATION dalam instalasi AC yang kesulitan dalam melakukan pengukuran ketinggian pada gedung bertingkat untuk dapat menentukan ukuran Ducting AC.

3.1.2 Analisis Sistem yang Akan Dibangun

Sistem yang akan dibangun adalah suatu aplikasi *mobile* menggunakan *platform android* dengan teknologi *accelerometer*. Tujuan yang ingin dicapai dari perancangan aplikasi ini adalah dapat mengukur ketinggian gedung agar dapat membantu CV. PROJECTRON CORPORATION dalam menentukan ukuran Ducting AC . Adapun gambaran umum tentang sistem yang akan dibangun dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Arsitektur Sistem yang Akan Dibangun.

Berikut penjelasan alur proses kerja sistem yang akan dibangun:

1. Pengguna mengaktifkan dan mengambil jarak titik koordinat GPS pada titik pengguna dengan titik objek.
2. Sistem mencari dan menyimpan nilai jarak titik koordinat GPS.
3. Setelah mendapatkan nilai jarak koordinat GPS, pengguna mengarahkan titik fokus bawah untuk mencari nilai tinggi h1 dan sudut kemiringan ponsel dengan memanfaatkan sensor *accelerometer* yang terdapat pada *smartphone android*.
4. Sistem menentukan nilai tinggi h1 dan nilai sudut kemiringan *accelerometer*.
5. Setelah didapat nilai tinggi h1, pengguna mengarahkan titik fokus atas dengan mengklik tombol hitung untuk mendapatkan hasil nilai tinggi objek.
6. Sistem memproses perhitungan dan menampilkan tinggi objek.

3.2 Analisis Metode

Analisis metode digunakan untuk mengetahui alur proses dari sebuah metode yang digunakan dapat diterpakan ke dalam aplikasi yang dibangun. Pembangunan aplikasi ini menggunakan metode *Trigonometri* dengan memanfaatkan *accelerometer* dan GPS, untuk mempermudah dalam perhitungan tinggi objek.

3.2.1 Analisis Metode Accelerometer

Accelerometer adalah sebuah sensor/tranduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. *Accelerometer* juga dapat digunakan untuk mengukur getaran yang terjadi pada kendaraan, bangunan, mesin, dan juga bisa digunakan untuk mengukur getaran yang terjadi di dalam bumi, getaran mesin, jarak yang dinamis, dan kecepatan dengan ataupun tanpa pengaruh gravitasi bumi.

Pada *smartphone*, *Accelerometer* merupakan sensor yang bisa membaca pergerakan sehingga dapat mengubah tampilan layar dari posisi *landscape* ke *portrait* atau sebaliknya dengan cukup memiringkan badan ponsel secara otomatis. Pada smartphone berbasis android terdapat sensor manager yang berfungsi untuk mengaktifkan sensor accelerometer dalam mencari nilai koordinat x,y,z dengan memiringkan handphone. Berikut nilai kemiringan sensor accelerometer.



Gambar 3. 2 Sudut kemirngan *Accelerometer*[16]

Tabel 3. 1 Nilai Accelerometer

Position	X	Y	Z
Vertical upright position	0.0	-1.0	0.0
Landscape Left	1.0	0.0	0.0
Landscape Right	-1.0	0.0	0.0
Upside Down	0.0	1	0.0
Flat Up	0.0	0.0	-1.0
Flat Down	0.0	0.0	1.0

3.2.2 Analisis Metode GPS

Global Positioning System (GPS) adalah suatu sistem navigasi atau penentu posisi berbasis satelit. Sistem ini didesain untuk memberikan posisi dan informasi mengenai waktu, secara kontinyu di seluruh dunia tanpa tergantung waktu dan cuaca. Penentuan posisi GPS digambarkan dengan menggunakan nilai koordinat X dan Y atau garis bujur dan garis lintang (*longitude/latitude*).

3.2.2.1 Titik Koordinat

Titik koordinat adalah titik yang berpedoman pada garis latitude dan longitude suatu daerah. Kaitannya dengan *latitude* dan *longitude* adalah, kedua garis lintang dan bujur inilah (*latitude* = garis lintang, *longitude* = garis bujur) yang menentukan di perolehnya suatu nilai derajat dari suatu titik yang diukur.

Koordinat pada umumnya dituliskan dengan format DD (*Degree Decimal*), terkadang juga dituliskan dengan format DMS (*Degree-Minute-Second*), jika berada dibelahan bumi utara dan timur maka bernilai positif, dalam hal ini untuk nilai *Lintang* atau *Latitude* dan *Bujur* atau *Longitude*.

3.2.2.2 Konversi Derajat Bujur dan Lintang

Perlu di ketahui pula konversi derajat menit (‘) dan detik (“) pada letak astronomis suatu tempat dalam satuan jarak (km dan m). Dalam pemakaian GPS sering didapatkan titik koordinat suatu posisi lokasi dalam bentuk *Latitude* (Lintang), *Longitude* (Bujur) atau *Degree* (Derajat). Dalam hal ini ada suatu formula yang digunakan untuk mengkonversi bilangan tersebut dari bentuk

Degree Minute Second (DMS) menjadi *Degree Decimal* (DD), atau sebaliknya konversi *Degree Decimal* (DD) *menjadi Degree Minute Second* (DMS).

a. Contoh perhitungan Konversi *Degree Decimal* (DD) menjadi *Degree Minute Second* (DMS):
   Bila diperoleh angka 109.036889 maka diubah menjadi 109˚.
   Angka dibelakang koma dikalikan dengan bilangan desimal 60, maka 0,036389 x 60 = 2,18334, urutan kedua yaitu 2 menit.
   Sisa angka dibelakang koma diatas (18334) dikalikan dengan bilangan desimal 60, maka 0,18334 x 60 = 11,0004, urutan ketiga yaitu 11 detik.
   Hasil dari konversi , secara lengkap ditulis 109˚2'11", hal ini juga berlaku pada Latitude.
b. Contoh perhitungan Konversi *Degree Minute Second* (DMS) menjadi *Degree Decimal* (DD):
   Bila diperoleh angka 7˚13'13", maka dapat diubah menjadi,
   = (Nilai Degree) + (Nilai Minute/60) + (Nilai Second/3600)
   = 7 + (13/60) + (13/3600)
   = 7.220278
   Maka hasil konversi DMS ke DD adalah 7.220278.

3.2.2.3 Menghitung Jarak Antar Koordinat

Jika menggunakan *Google-Earth* akan langsung dapat diketahui jarak antar titik-titik koordinat dengan menarik garis dari satu sisi titik ke titik yang diinginkan. Jika tidak maka bisa dihitung menggunakan rumus 1' = 1.825 m, 1" = 30,416 m. Selanjutnya selisih angka N menghasilkan sumbu vertical, sedangkan selisih angka E menghasilkan sumbu horizontal. Berikut adalah objek yang djadikan ilustrasi mencari nilai jarak anatara 2 titik koordinat adalah kantor CV. PROJECTRON CORPORATION.

Diketahui nilai koordniat DMS 1, S -7˚1'41.9952" dan E 107˚31'18.1128" sedangkan nilai koordinat DMS 2, S -7˚1'42.006" dan E 107˚31'18.354"

Tabel 3. 2 Perhitungan Jarak Koordinat

	Titik 1	**Titik 2**	**Selisih**	**Dikalikan 30.416**
S	-7˚1'41.9952"	-7˚1'42.006"	- 0.0108	- 0.3284928
E	107˚31'18.1128	107˚31'18.354"	- 0.2412	- 7.3363392

Maka jarak titik 1 dengan titik 2 yaitu:

$= \sqrt{((-0.3284928)^2 + (-7.3363392)^2)\text{m}^2}$

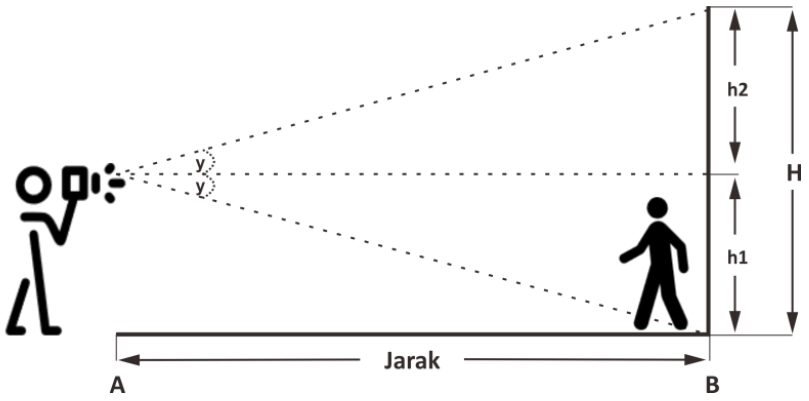
$= \sqrt{(0.1079 + 53.8218)\text{m}^2}$

$= \sqrt{53.929\ \text{m}^2}$

= 7,343 m

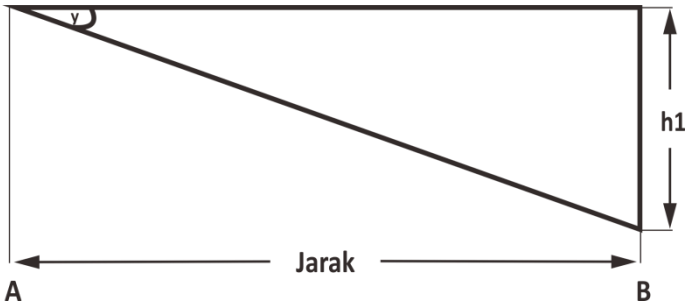
3.2.3 Analisis Metode Trigonometri

Analisi metode trigonometri pada aplikasi pengukuran tinggi objek yaitu dengan memanfaatkan sensor *accelerometer* dan fitur GPS pada *android*. Untuk mendapatkan nilai tinggi objek, pengguna harus mencari sudut kemiringan ponsel dengan memanfaatkan sensor *accelerometer* dan mencari nilai jarak kedua koordinat GPS menggunakan fitur GPS pada *smatphone android*.



Gambar 3. 3 Ilustrasi Pengukuran Tinggi Objek

Setelah didapat nilai jarak antara titik 1 dan 2 dari perhitungan koordinat pada tabel 3.2. maka dapat dicari nilai h1 dari perhitungan jarak kedua koordinat GPS dan nilai sudut *accelerometer*, seperti pada gambar dibawah.



Gambar 3. 4 Perhitungan Nilai Tinggi h1

Dengan rumus sebagai berikut:

Jarak = tan y * h1

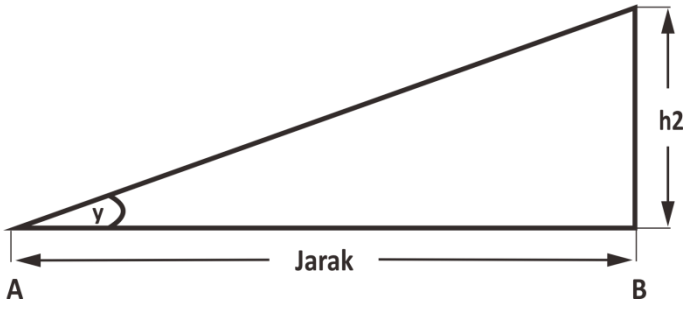
h1 = Jarak / tan y

= 7.343 / 3.077

h1 = 2.4

Nilai tan y, di dapat dari kemiringan *accelerometer*.

Setelah didapat nilai h1, maka dapat dicari nilai h2 dari perhitungan jarak kedua koordinat GPS dan nilai sudut *accelerometer* untuk mendapat nilai tinggi objek, seperti pada gambar dibawah.



Gambar 3. 5 Perhitungan Nilai Tinggi h2

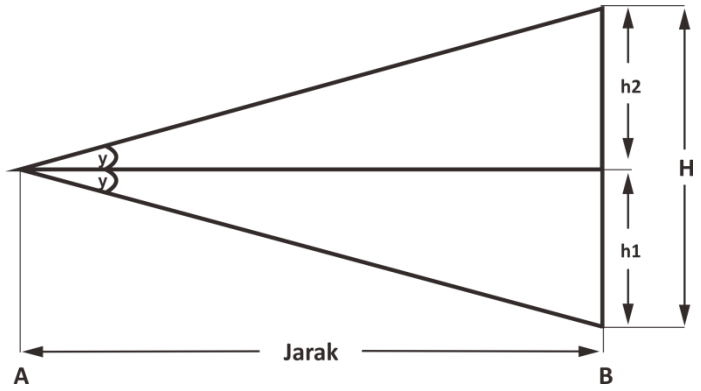
Dengan rumus sebagai berikut:

h2 = tan y * Jarak, maka

h2 = 0.69 * 7.343

= 5.06

Setelah didapat nilai h1 dan h2, maka dapat dihitung nilai tinggi total dari objek, terlihat pada gambar sebagai berikut:



Gambar 3. 6 Perhitungan Tinggi Total

Dengan rumus, H = h1 + h2

H = 2.4 + 5.06

= 7.46

3.3 Analisis Kebutuhan Non–Fungsional

Adapun kebutuhan non-fungsional untuk menjalankan aplikasi ini meliputi kebutuhan perangkat keras, kebutuhan perangkat lunak dan pengguna sistem yang akan memakai aplikasi. Analisis kebutuhan non-fungsional bertujuan agar aplikasi yang dibangun dapat digunakan sesuai dengan kebutuhan pengguna aplikasi dalam mencari informasi yang dibutuhkan.

3.3.1 Analisis Kebutuhan Perangkat Keras

Kebutuhan perangkat keras untuk menjalankan aplikasi ini adalah *smartphone Android* yang memiliki beberapa spesifikasi utama sebagai berikut:

1. Sistem operasi Android v4.4.2 (*Kitkat*).

2. Kamera 5 MP, untuk keperluan pengambilan gambar yang dilakukan secara langsung.
3. *Random Access Memory* (RAM) 1 GB.
4. Memiliki fiture GPS.
5. Sensor *Accelerometer*.

3.3.2 Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dipakai dalam pembangunan aplikasi, diantaranya adalah sebagai berikut:

1. Sistem Operasi Windows 7 Ultimate.
2. *Java Development Kit* versi 7, untuk keperluan kompilasi kode-kode program.
3. *Microsoft Net Framework* 3.5 dan 4.5.
4. *Basic4Android*, digunakan sebagai perangkat kompilasi kode-kode program.
5. *Android SDK*, untuk pengambilan aplikasi *android*.
6. *Android Development Tool*, sebagai plugin untuk mengintegrasikan android SDK pada *Basic4Android*.

3.3.3 Analisis Kebutuhan User

Kebutuhan pengguna yang harus dimiliki agar tidak kesulitan dalam menjalankan aplikasi, diantaranya adalah sebagai berikut:

1. Memahami operasional *Smartphone*.
2. Dapat mengoperasikan Aplikasi pada *Smartphone*.
3. Minimal memiliki *Smartphone Android*.

3.4 Analisis Kebutuhan Fungsional

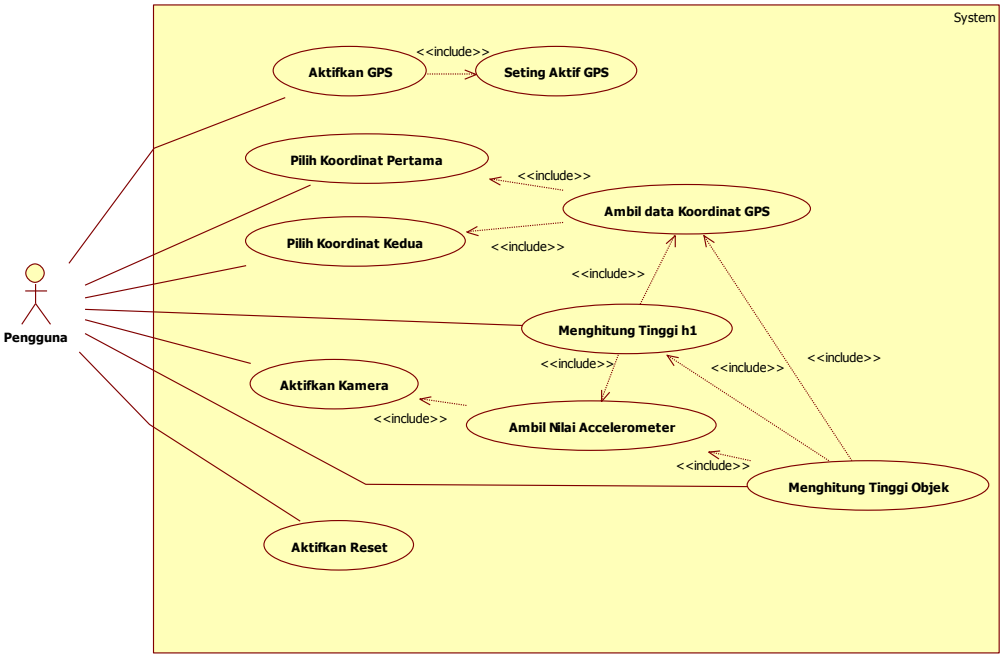
Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar dapat berjalan dengan baik sesuai kebutuhan.

Analisis yang dilakukan dimodelkan dengan UML (*Unfield Modeling Language*). Tahap-tahap pemodelan dalam analisis tersebut antaranya diagram *use*

case, *diagram activity*, *diagram class* dan *diagram sequence*. Diagram tersebut dipergunakan sebagai desain rancangan untuk mempermudah proses pembangunan aplikasi khususnya untuk aplikasi yang dikembangkan menggunakan pemrograman berorientasi objek.

3.4.1 Use Case Diagram

Use case diagram digunakan untuk mengetahui apa saja yang dapat dilakukan oleh pengguna atau aktor terhadap fungsionalitas yang terdapat pada aplikasi yang dirancang.



Gambar 3. 7 Use Case Diagram

3.4.1.1 Identifikasi Actor

Aktor adalah abstraksi dari orang dan sistem yang lain yang mengaktifkan fungsi dari target sistem. Berikut adalah aktor yang berperan dalam menjalankan sistem yang dibangun.

Tabel 3. 3 Aktor *Use Case*

No	Aktor	Deskripsi
A-01	User	Merupakan aktor dari aplikasi yang dibangun atau pengguna aplikasi dari pengukur ketinggian suatu objek.

Penggambaran *use case* diagram sistem yaitu *use case* menunjukkan bagaimana interaksi antara aktor dan sistem. Aktor user yaitu yang menjalankan aplikasi.

3.4.1.2 Identifikasi Use Case

Berikut tabel Identifikasi *Use Case* yang terdapat pada aplikasi:

Tabel 3. 4 *Use Case Aktor*

Use Case Aplikasi User		
No	**Use Case**	**Deskripsi**
UC-01	Aktifkan GPS	Fungsional untuk mengaktifkan fitur GPS pada *Smartphone Android*.
UC-02	Pilih Koordinat Pertama	Fungsional untuk mengambil nilai jarak koordinat GPS pertama.
UC-03	Pilih Koordinat Kedua	Fungsional untuk mengambil nilai jarak koordinat GPS kedua.
UC-04	Aktifkan Kamera	Fungsional untuk membidik objek serta menentukan nilai kemiringan dari *accelerometer*.
UC-05	Menghitung Tinggi h1	Fungsional untuk mendapatkan nilai tinggi h1
UC-06	Menghitung Tinggi Objek	Fungsional untuk mendapatkan nilai tinggi total dari suatu objek.
UC-07	Aktifkan Reset	Fungsional untuk mereset atau menghapus data nilai GPS, sudut kemiringan *accelerometer*, nilai tinggi h1,dan nilai tinggi objek.

3.4.2 Skenario Use Case

Skenario *use case* mendeskripsikan urutan langkah-langkah dalam proses bisnis, baik dilakukan aktor terhadap sistem maupun yang dilakukan oleh sistem terhadap aktor.

3.4.2.1 Skenario Use Case Aktifkan GPS

Skenario *use case aktifkan GPS* dapat dilihat pada tabel 3.5.

Tabel 3. 5 Skenario *Use Case* Aktifkan GPS

Identifikasi	
Nama Use Case User	Aktifkan GPS
Aktor	*Pengguna*
Tujuan	Digunakan untuk mengaktifkan fitur GPS pada *smartphone android*
Keadaan Awal	Sistem menampilkan layar utama aplikasi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan layar utama aplikasi
2. Pilih tombol GPS untuk mengaktifkan fitur GPS.	
	3. Menampilkan Setting Aktif GPS
4. Pilih tombol aktif GPS	
	5. Mengaktifkan GPS

3.4.2.2 Skenario Use Case Pilih Koordinat Pertama

Skenario *use case pilih koordinat pertama* dapat dilihat pada tabel 3.6.

Tabel 3. 6 Skenario *Use Case* Aktifkan Koordinat Pertama

Identifikasi	
Nama Use Case User	Pilih Koordinat Pertama
Aktor	*Pengguna*
Tujuan	Digunakan untuk mengambil nilai jarak koordinat GPS pertama
Keadaan Awal	Sistem menampilkan layar utama aplikasi

Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pilih tombol Koor 1 untuk mengambil nilai jarak dari koordinat GPS pertama.	
	2. Sistem mencari nilai koordinat GPS
	3. Sistem menyimpan nilai koordinat GPS
	4. Sistem menampilkan nilai Koordinat GPS yang digunakan sebagai nilai jarak koordinat GPS pertama

3.4.2.3 Skenario Use Case Pilih Koordinat Kedua

Skenario *use case pilih koordinat kedua* dapat dilihat pada tabel 3.7.

Tabel 3. 7 Skenario *Use Case* Pilih Koordinat Kedua

Identifikasi	
Nama Use Case User	Pilih Koordinat Kedua
Aktor	*Pengguna*
Tujuan	Digunakan untuk mengambil nilai jarak koordinat GPS kedua
Keadaan Awal	Sistem menampilkan layar utama aplikasi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pilih tombol Koor 2 untuk mengambil nilai jarak dari koordinat GPS kedua.	
	2. Sistem mencari nilai koordinat GPS
	3. Sistem menyimpan nilai koordinat GPS
	4. Sistem menampilkan nilai koordinat GPS yang digunakan sebgai nilai jarak koordinat GPS kedua

3.4.2.4 Skenario Use Case Aktifkan Kamera

Skenario *use case aktifkan kamera* dapat dilihat pada tabel 3.8.

Tabel 3. 8 Skenario *Use Case* Aktifkan Kamera

Identifikasi	
Nama Use Case User	Aktifkan Kamera
Aktor	*Pengguna*
Tujuan	Digunakan untuk membidik objek serta untuk mendapatkan nilai kemiringan dari sensor *accelerometer*
Keadaan Awal	Sistem menampilkan kamera
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Menampilkan kamera
2. Membidik objek yang akan diukur dan mencari nilai sudut kemiringan	
	3. Sistem mencari nilai sudut kemiringan dari *accelerometer*
	4. Sistem menyimpan nilai sudut kemiringan dari *accelerometer*
	5. Sistem menampilkan nilai sudut kemringan dari *accelerometer*

3.4.2.5 Skenario Use Case Menghitung Tinggi h1

Skenario *use case menghitung tinggi objek* dapat dilihat pada tabel 3.9

Tabel 3. 9 Skenario *Use Case* Menghitung Tinggi Objek

Identifikasi	
Nama Use Case User	Menghitung Tinggi h1
Aktor	*Pengguna*
Tujuan	Digunakan untuk mendapatkan nilai tinggi h1 dari perhitungan nilai jarak koordinat GPS dan nilai *accelerometer*
Keadaan Awal	Sistem menampilkan layar utama aplikasi

Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pilih tombol h1 untuk mendapatkan nilai tinggi h1 dari sebuah objek	
	2. Sistem mengambil nilai jarak koordinat GPS pertama
	3. Sistem mengambil nilai jarak koordinat GPS kedua
	4. Sistem mengambil nilai kemiringan dari *accelerometer*
	5. Sistem memproses perhitungan tinggi h1
	6. Mendapatkan nilai tinggi h1
	7. Menampilkan nilai tinggi h1

3.4.2.6 Skenario Use Case Menghitung Tinggi Objek

Skenario *use case menghitung tinggi objek* dapat dilihat pada tabel 3.10.

Tabel 3. 10 Skenario *Use Case* Menghitung Tinggi Objek

Identifikasi	
Nama Use Case User	Menghitung Tinggi Objek
Aktor	*Pengguna*
Tujuan	Digunakan untuk mendapatkan nilai tinggi objek dari perhitungan nilai jarak koordinat GPS dan nilai *accelerometer*
Keadaan Awal	Sistem menampilkan layar utama aplikasi
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
1. Pilih tombol hitung untuk mendapatkan nilai tinggi dari sebuah objek	
	2. Sistem mengambil nilai jarak koordinat GPS pertama
	3. Sistem mengambil nilai jarak koordinat GPS kedua
	4. Sistem mengambil nilai kemiringan dari *accelerometer*

	5. Sistem mengambil nilai tinggi h1
	6. Sistem memproses perhitungan tinggi objek
	7. Mendapatkan nilai tinggi objek
	8. Menampilkan nilai tinggi objek

3.4.2.7 Skenario Use Case Aktifkan Reset

Skenario *use case aktifkan reset* dapat dilihat pada tabel 3.11.

Tabel 3. 11 Skenario *Use Case* Aktifkan Reset

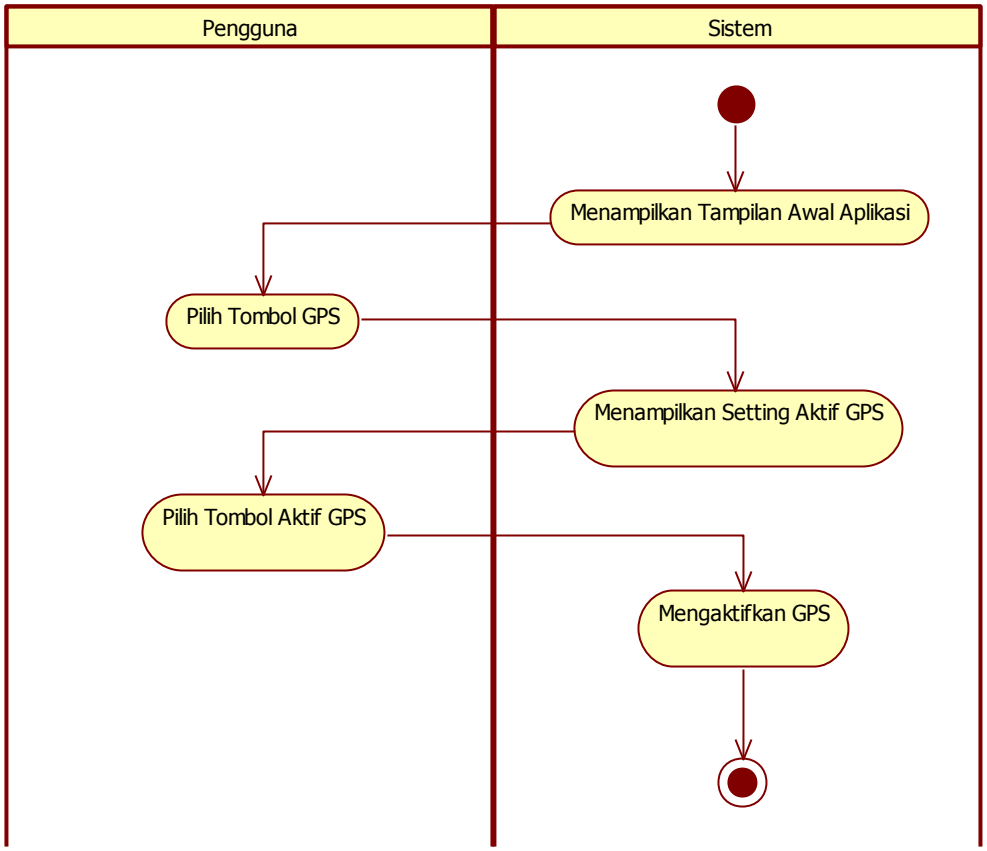
Identifikasi	
Nama Use Case User	Aktifkan Reset
Aktor	*Pengguna*
Tujuan	Digunakan untuk mereset atau menghapus semua data nilai GPS, sudut kemiringan *accelerometer*, nilai tinggi h1, dan nilai tinggi objek.
Keadaan Awal	Sistem menampilkan kamera
Skenario Utama	
Aksi Aktor	**Reaksi Sistem**
	1. Sistem menampilkan layar utama aplikasi.
2. Pilih tombol reset untuk mereset atau menghapus semua data nilai GPS, sudut kemiringan *accelerometer*, nilai tinggi h1, dan nilai tinggi objek.	
	3. Sistem menghapus semua data nilai GPS, sudut kemitringan *accelerometer*, nilai tinggi h1, dan nilai tinggi objek.

3.4.3 Activity Diagram

Activity diagram digunakan untuk menggambarkan alur rangkaian proses yang terjadi di dalam sistem aplikasi serta menjelaskan bagaimana sebuah *use case* terjadi atau dipicu dari *use case* yang lain.

3.4.3.1 Activity Diagram Menu Aktifkan GPS

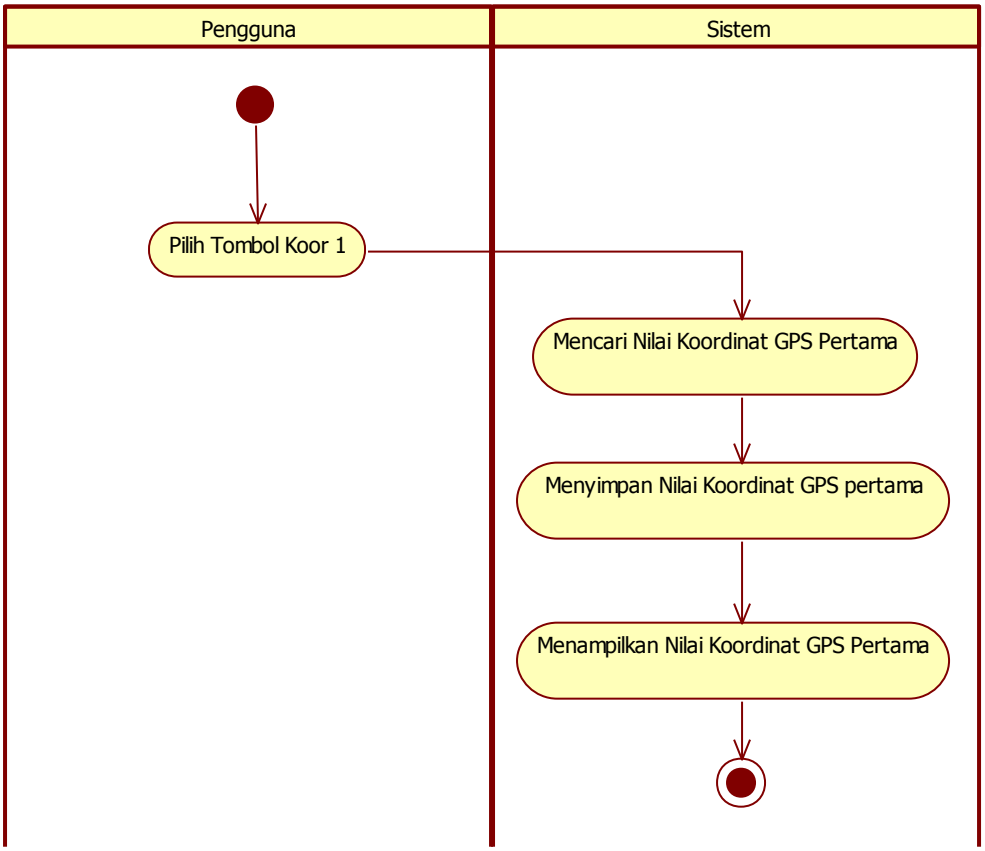
Activity diagram *menu aktifkan GPS* menggambarkan alur aktifitas untuk memulai aplikasi menu aktif GPS antara *user* dan sistem seperti terlihat pada gambar 3.8.



Gambar 3. 8 *Activity Diagram* Menu Aktifkan GPS

3.4.3.2 Activity Diagram Pilih Koordinat Pertama

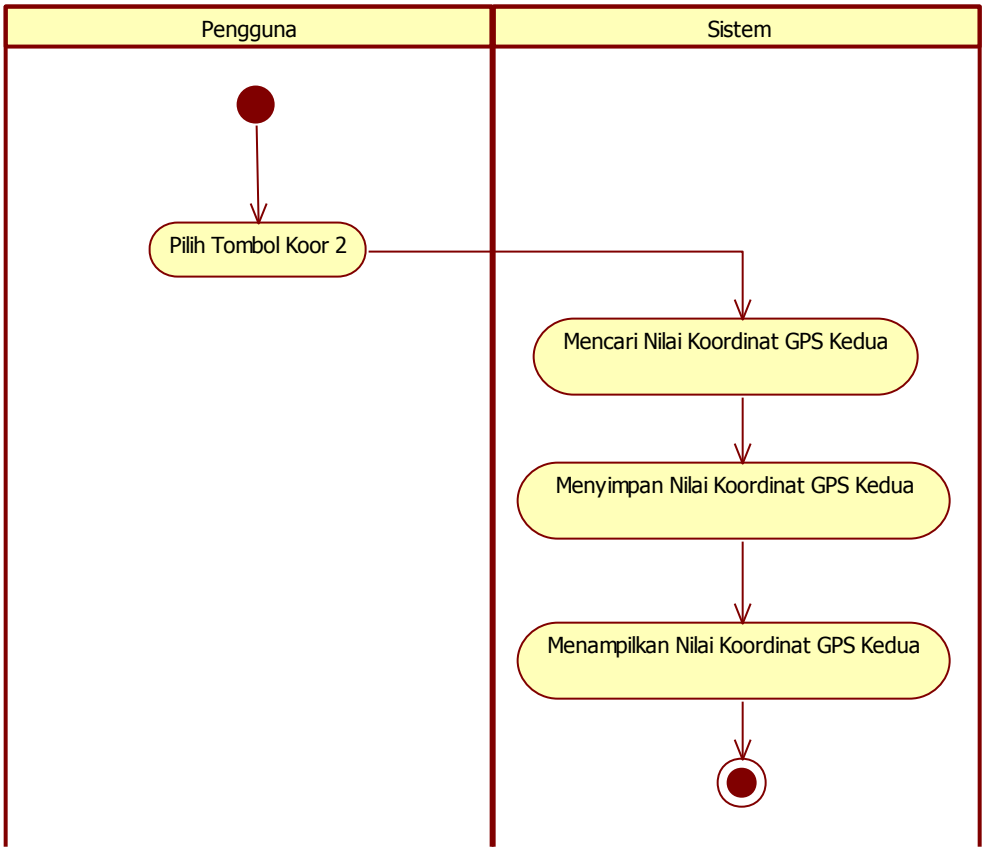
Activity diagram *pilih koordinat pertama* menggambarkan alur aktifitas untuk memulai aplikasi menu koordinat pertama antara *user* dan sistem seperti terlihat pada gambar 3.9.



Gambar 3. 9 *Activity Diagram* Pilih Koordinat GPS Pertama

3.4.3.3 Activity Diagram Pilih Koordniat Kedua

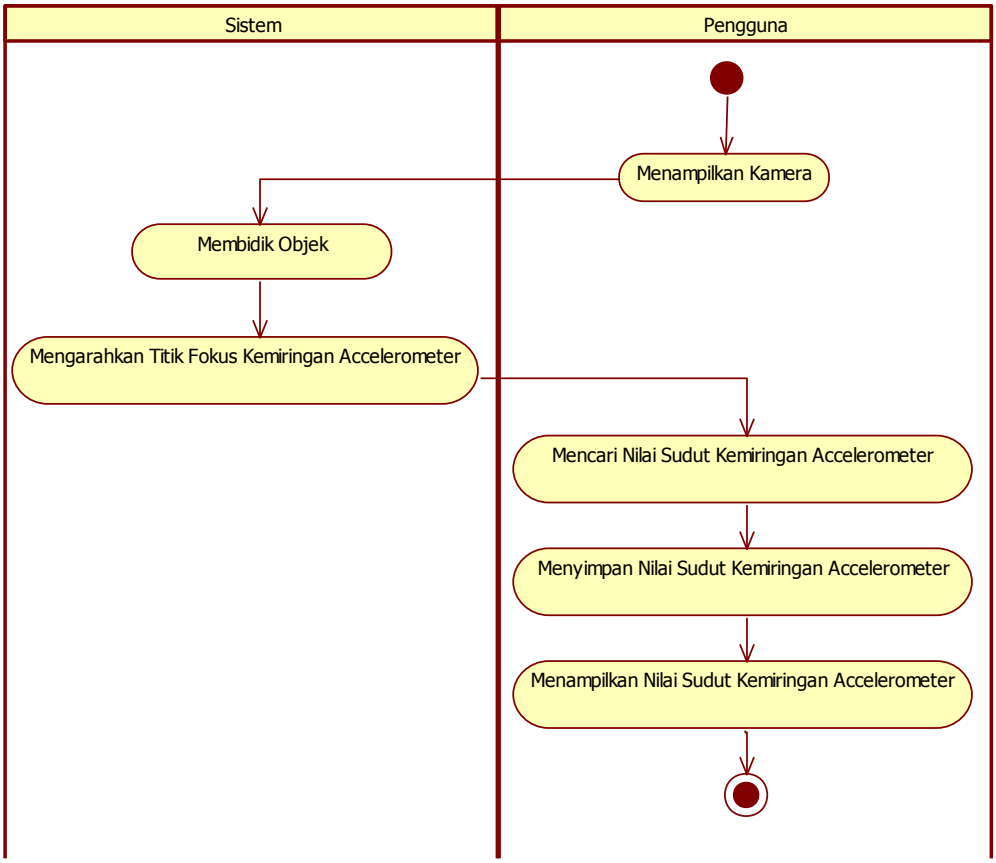
Activity diagram *pilih koordinat kedua* menggambarkan alur aktifitas untuk memulai aplikasi menu koordinat kedua antara *user* dan sistem seperti terlihat pada gambar 3.10.



Gambar 3. 10 *Activity Diagram* Pilih Koordinat GPS Kedua

3.4.3.4 Activity Diagram Aktifkan Kamera

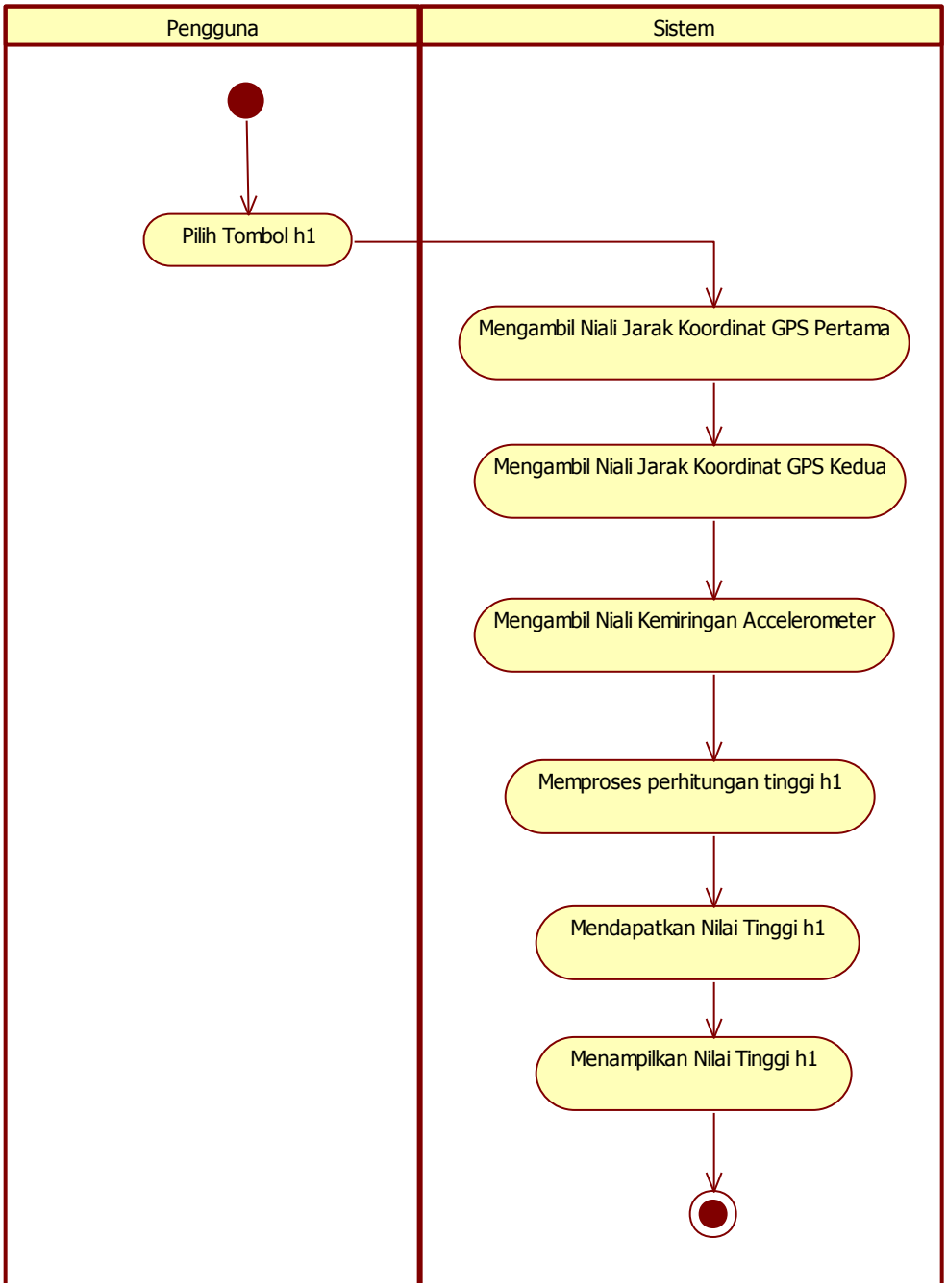
Activity diagram *aktifkan kamera* menggambarkan alur aktifitas untuk memulai aplikasi kamera antara *user* dan sistem seperti terlihat pada gambar 3.11.



Gambar 3. 11 *Activity Diagram* Aktifkan Kamera

3.4.3.5 Activity Diagram Menghitung Tinggi h1

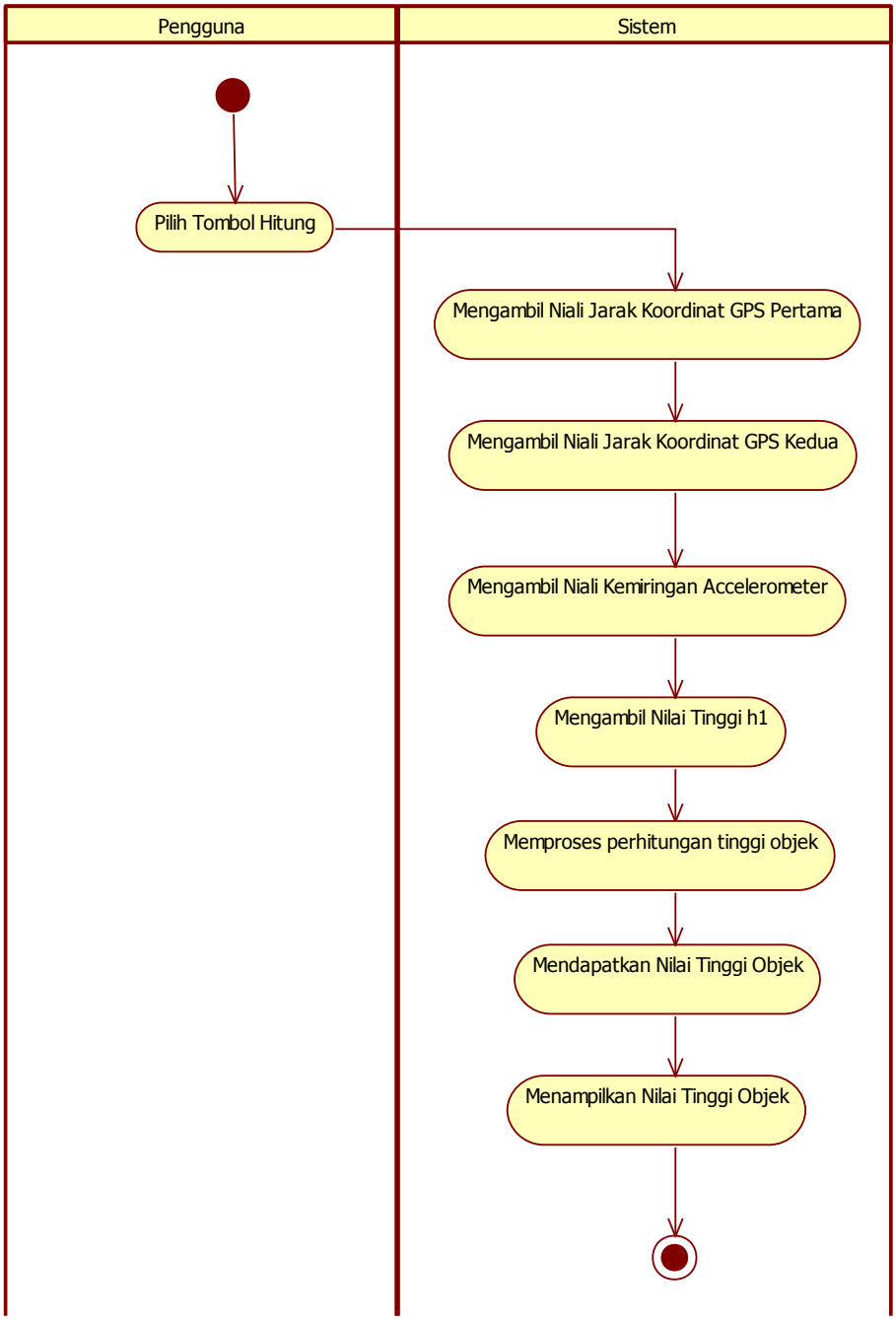
Activity diagram *menghitung tinggi h1* menggambarkan alur aktifitas untuk memulai menghitung tinggi h1 antara *user* dan sistem seperti terlihat pada gambar 3.12.



Gambar 3. 12 *Activity Diagram* Menghitung Tinggi h1

3.4.3.6 Activity Diagram Menghitung Tinggi Objek

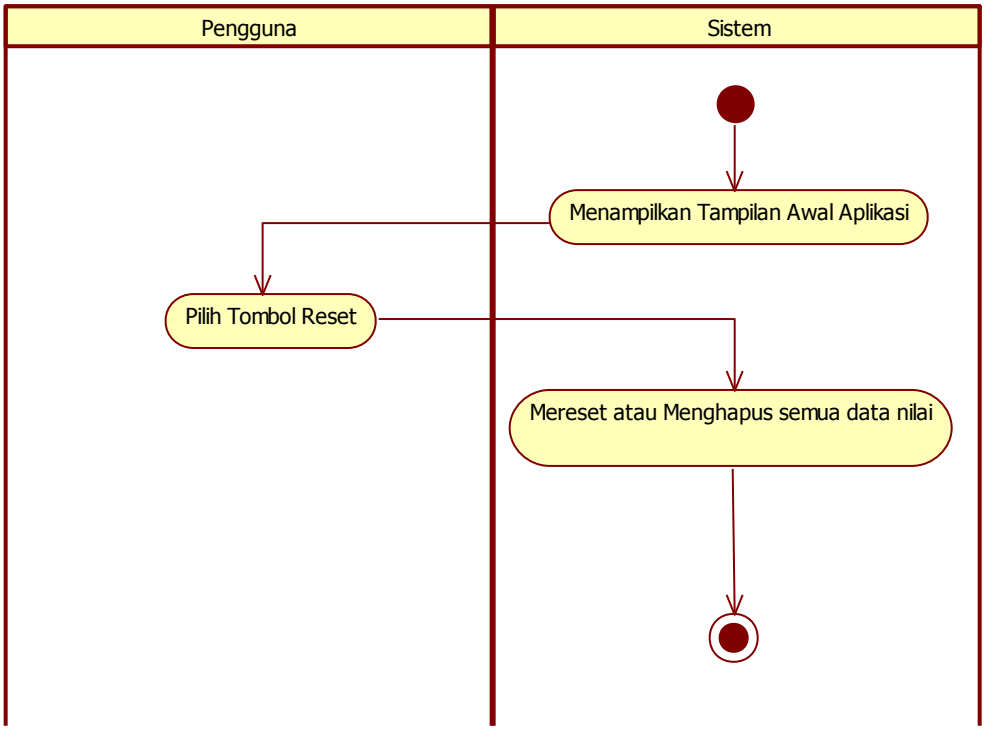
Activity diagram *menghitung tinggi objek* menggambarkan alur aktifitas untuk memulai menghitung tinggi objek antara *user* dan sistem seperti terlihat pada gambar 3.13.



Gambar 3. 13 *Activity Diagram* Menghitung Tinggi Objek

3.4.3.7 Activity Diagram Aktifkan Reset

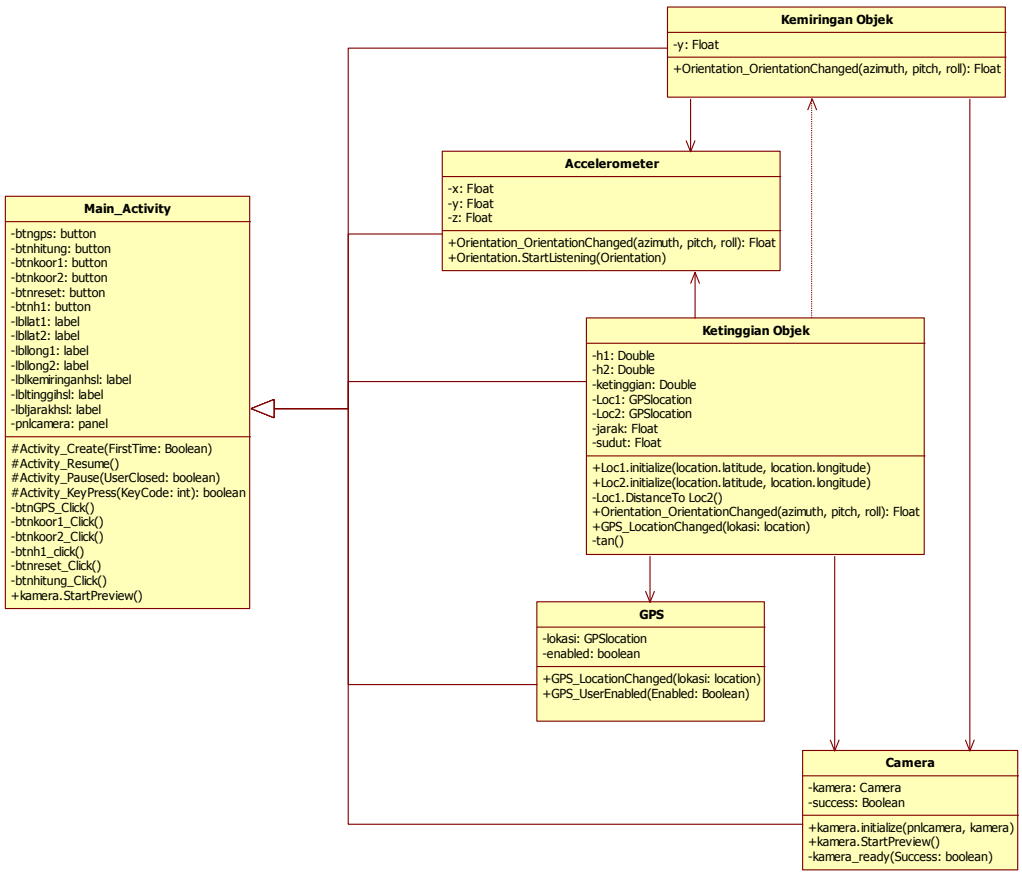
Activity diagram *aktifkan reset* menggambarkan alur aktifitas untuk mereset atau menghapus semua data nilai antara *user* dan sistem seperti terlihat pada gambar 3.14.



Gambar 3. 14 *Activity Diagram* Aktifkan Reset

3.4.4 Class Diagram

Class Diagram menggambarkan struktur dan hubungan antar objek-objekyang ada pada sistem. Struktur itu meliputi atribut-atribut dan metode metode yang ada pada masing-masing class. Hubungannya dari masing-masing class yang ada tersebut digambarkan seperti pada gambar dibawah.

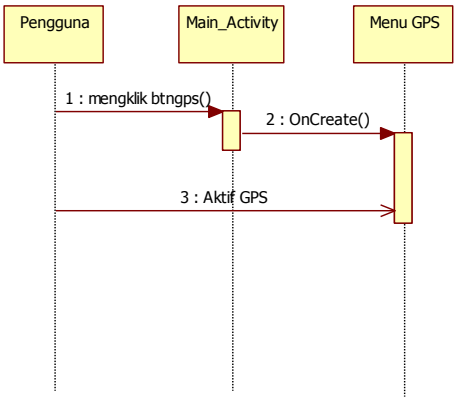


Gambar 3. 15 *Class Diagram*

3.4.5 Sequence Diagram

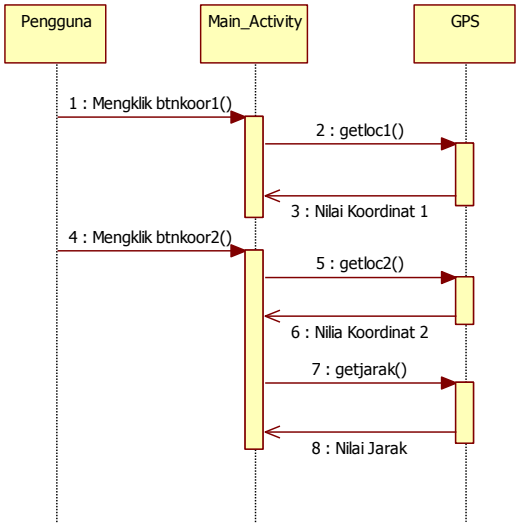
Diagram *sequence* menggambarkan urutan aktifitas yang terjadi di dalam sistem. Diagram ini menunjukkan pengiriman pesan-pesan yang melewati objek yang terlibat di dalam sistem serta tahapan langkah yang harus dilalui untuk mencapai *use case* tertentu.

1. *Diagram Sequence* Menu Aktif GPS



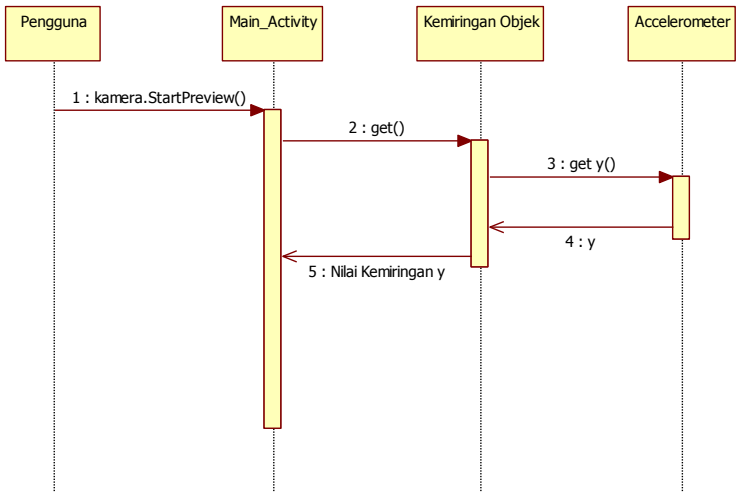
Gambar 3. 16 *Sequence Diagram* Menu Aktif GPS

2. *Diagram Sequence* Mencari Nilia Koordinat GPS sebagai Jarak



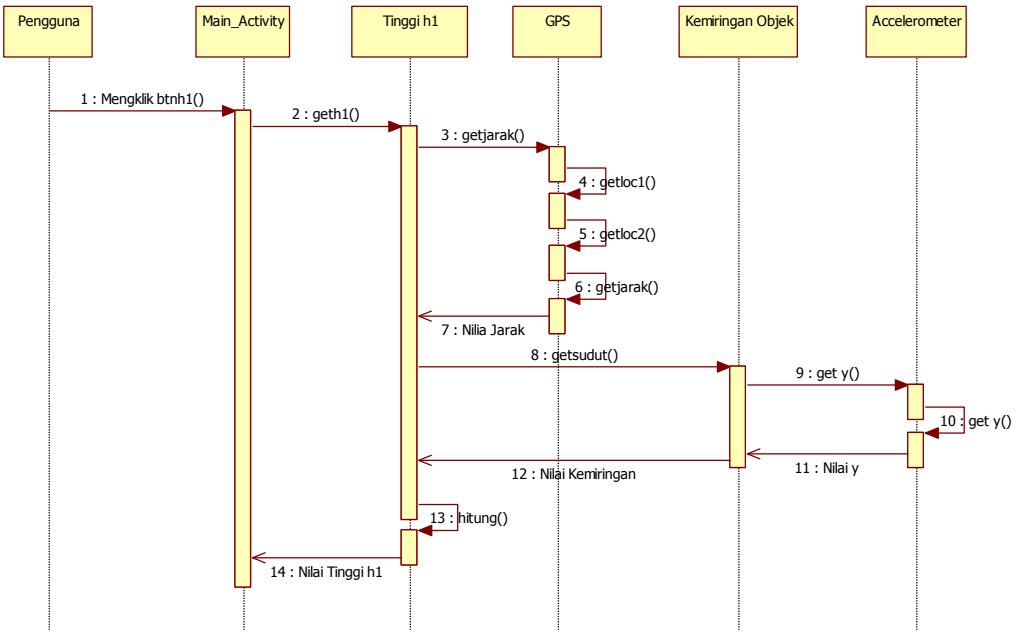
Gambar 3. 17 *Sequence Diagram* Mencari Nilia Koordinat GPS sebagai Jarak

3. *Diagram Sequence* Mencari Nilia Kemiringan dari Accelerometer



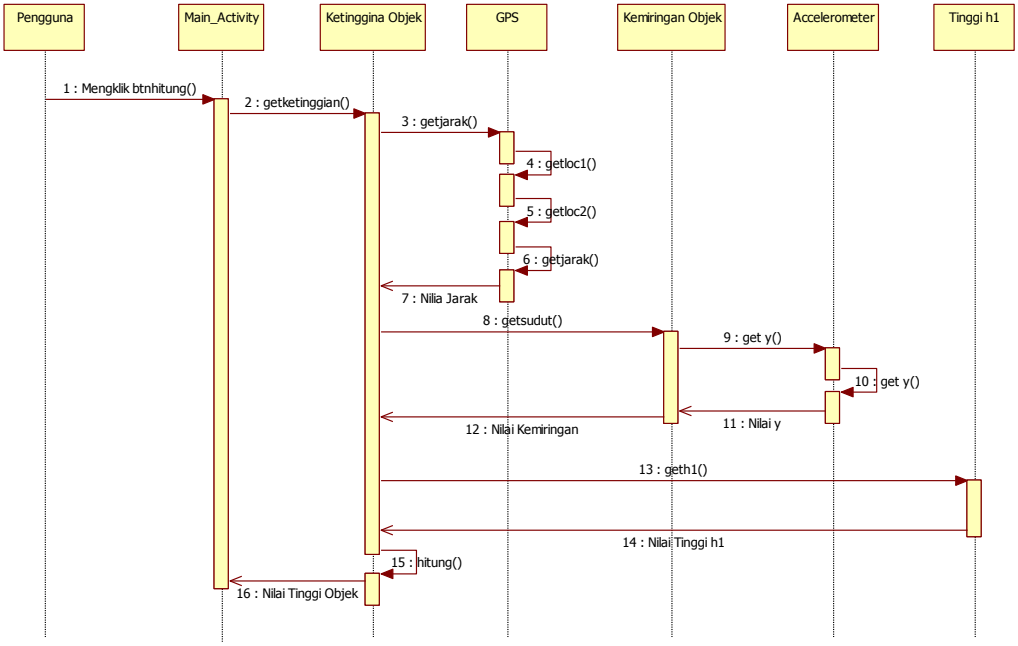
Gambar 3. 18 *Sequence Diagram* Mencari Nilia Kemiringan dari *Accelerometer*

4. *Diagram Sequence* Menghitung Tinggi h1



Gambar 3. 19 *Sequence Diagram* Menghitung Tinggi Objek

5. *Diagram Sequence* Menghitung Tinggi Objek



Gambar 3. 20 *Sequence Diagram* Menghitung Tinggi Objek

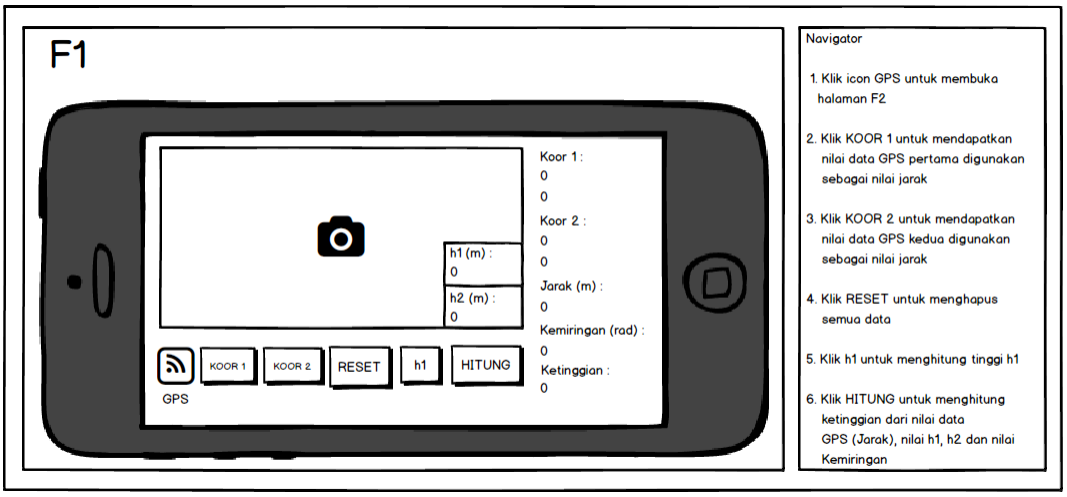
3.5 Perancangan Sistem

Adapun tahapan yang dilakukan dalam perancangan sistem ini membahas mengenai tujuan perancangan antarmuka, perancangan jaringan simantik, dan perancangan prosedural.

3.5.1 Perancangan Antarmuka Sistem

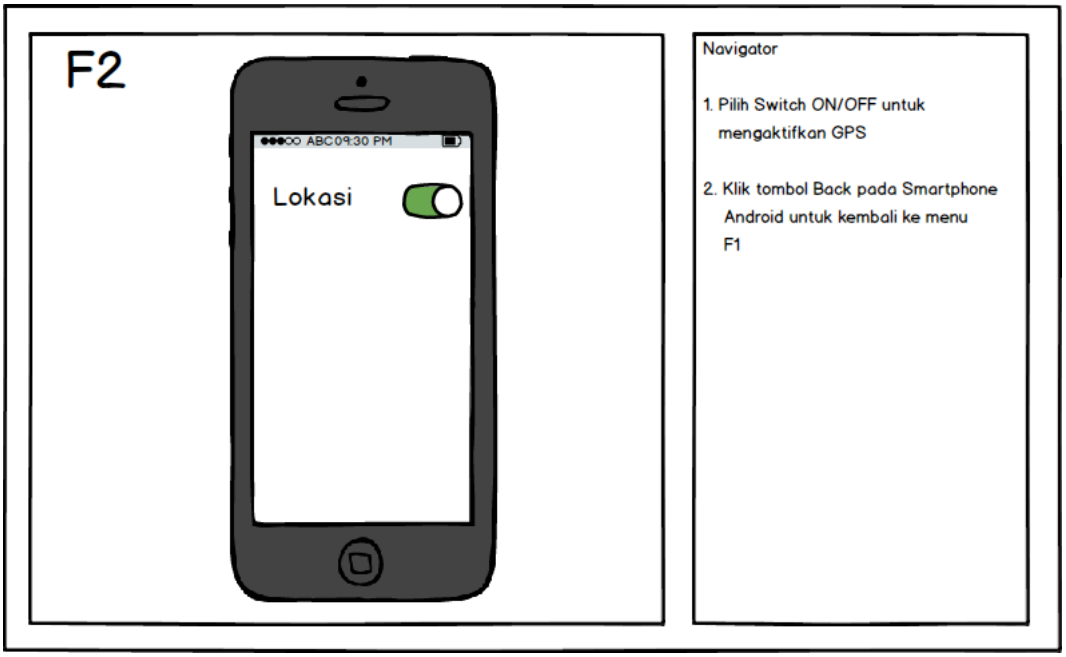
Tahap perancangan antarmuka bertujuan untuk memberikan gambaran tentang aplikasi yang akan dibangun, sehingga akan mempermudah dalam mengimplementasikan aplikasi serta akan memudahkan pembuatan aplikasi. Berikut ini gambaran perancangan antarmuka yang ada dari aplikasi yang akan dibangun yaitu:

3.5.1.1 Perancangan Antarmuka Halaman Utama



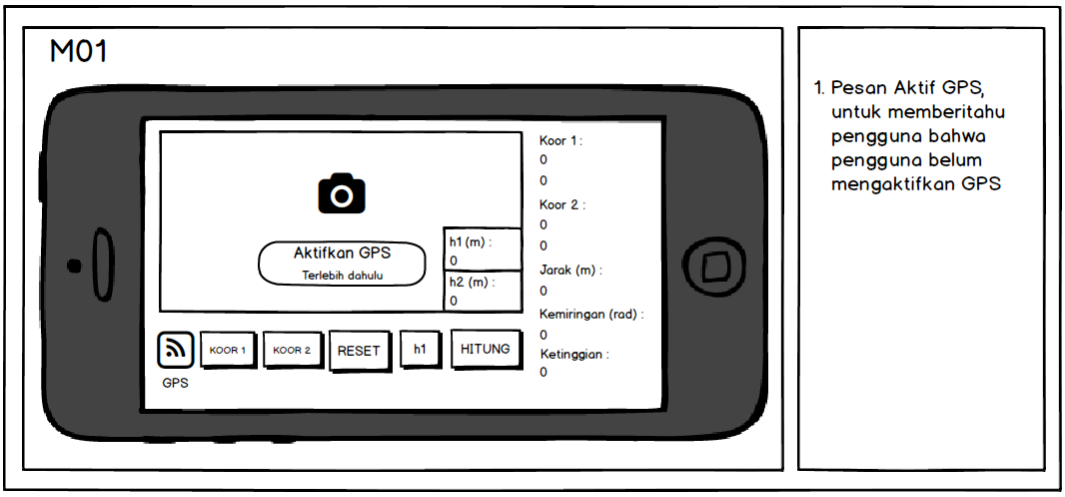
Gambar 3. 21 Perancangan Antarmuka Halaman Utama

3.5.1.2 Perancangan Antarmuka Menu Aktif GPS



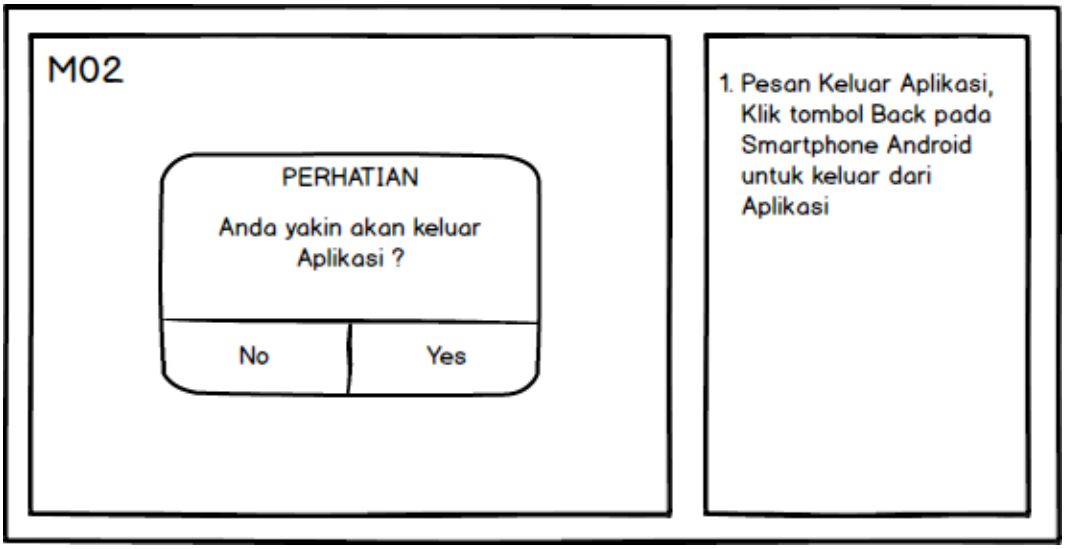
Gambar 3. 22 Perancangan Antarmuka Menu Aktif GPS

3.5.1.3 Perancangan Pesan Aktif GPS



Gambar 3. 23 Perancangan Antarmuka Pesan Aktif GPS

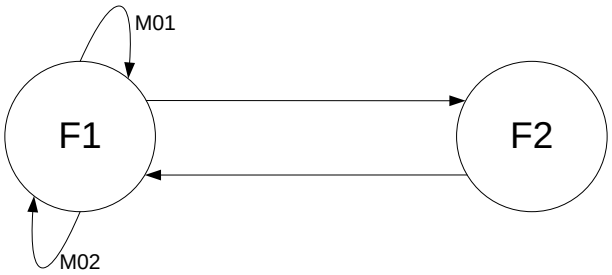
3.5.1.4 Perancangan Pesan Keluar Aplikasi



Gambar 3. 24 Perancangan Antarmuka Pesan Keluar Aplikasi

3.5.2 Perancangan Jaringan Semantik

Jaringan semantik merupakan gambaran grafis yang menunjukkan antar objek (lingkaran) dan garis yang menggambarkan formasi antar objek tersebut. Adapun jaringan semantik untuk aplikasi pengukur ketinggian dan kemiringan objek, dapat dilihat pada gambar dibawah:



Gambar 3. 25 Jaringan Semantik

Keterangan:

F1 : Tampilan Halaman Utama

F2 : Tampilan Menu Aktif GPS

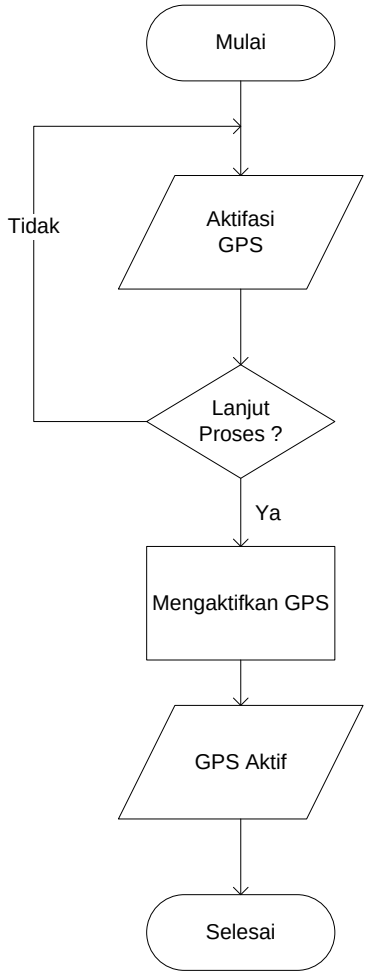
M01 : Tampilan Pesan Aktifkan GPS

M02 : Tampilan Pesan Keluar Aplikasi

3.5.3 Perancangan Prosedural

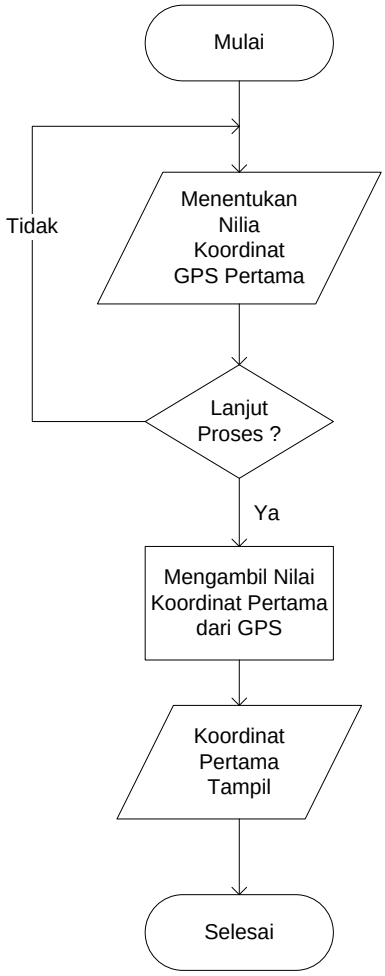
Perancangan prosedural digunakan untuk menggambarkan langkah-langkah atau alur dari setiap proses yang ada pada sistem. Berikut adalah rancangan prosedural dari aplikasi yang akan dibangun.

3.5.3.1 Perancangan Prosedural Aktif GPS



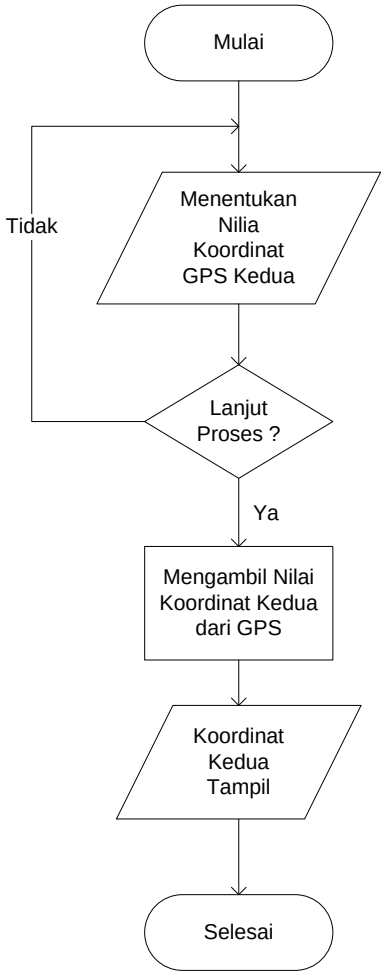
Gambar 3. 26 Prosedural Aktif GPS

3.5.3.2 Perancangan Prosedural Koordinat Pertama



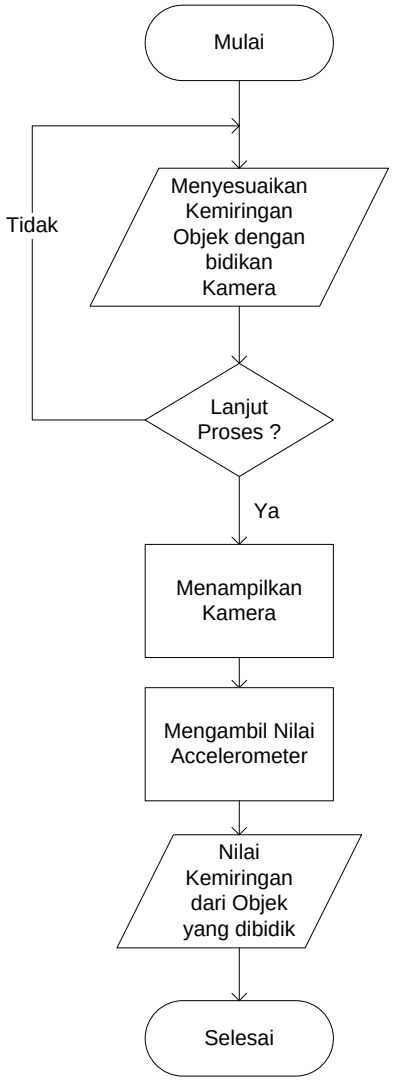
Gambar 3. 27 Prosedural Koordinat Pertama

3.5.3.3 Perancangan Prosedural Koordinat Kedua



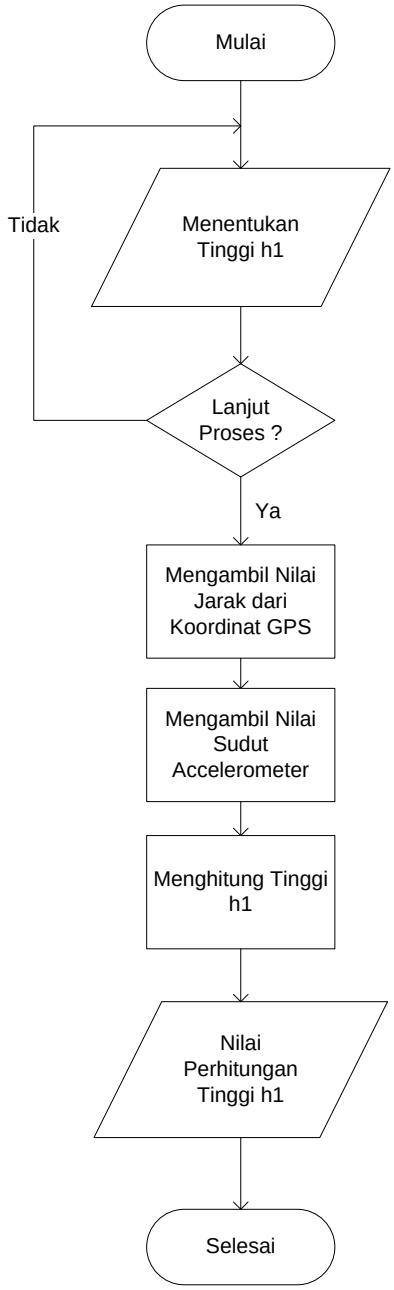
Gambar 3. 28 Prosedural Koordinat Kedua

3.5.3.4 Perancangan Prosedural Mentukan Kemiringan dengan Kamera



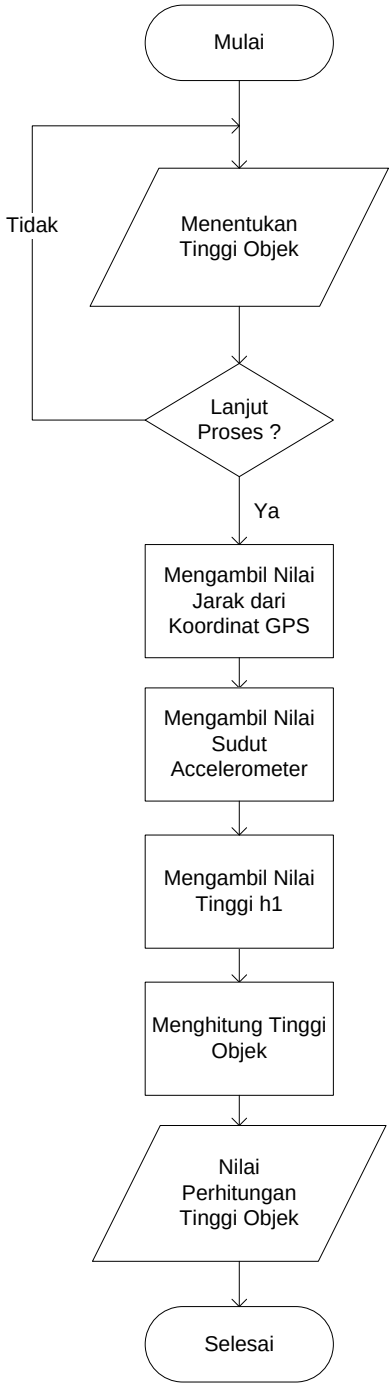
Gambar 3. 29 Prosedural Menentukan Kemiringan dengan Kamera

3.5.3.5 Perancangan Prosedural Menghitung Tinggi h1



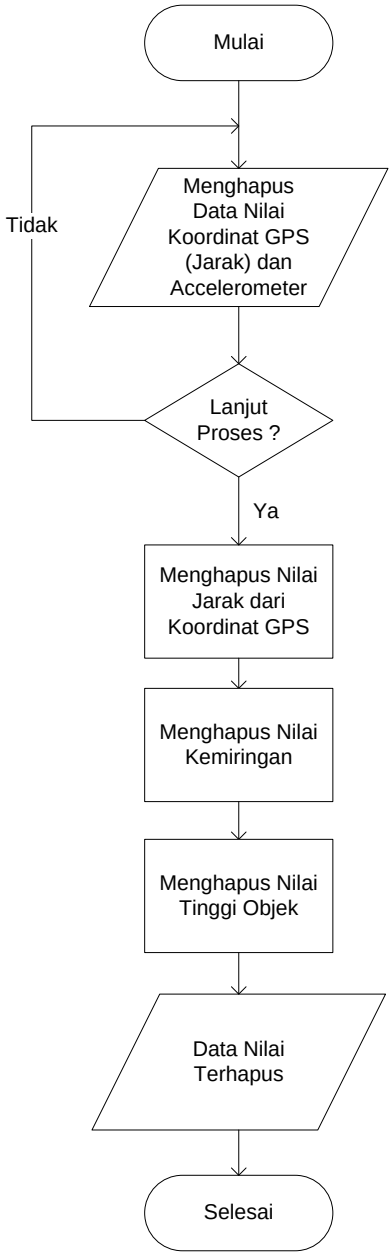
Gambar 3. 30 Prosedural Menghitung Tinggi h1

3.5.3.6 Perancangan Prosedural Menghitung Tinggi Objek



Gambar 3. 31 Prosedural Menghitung Tinggi Objek

3.5.3.7 Perancangan Prosedural Aktif Reset



Gambar 3. 32 Prosedural Aktif Reset