

## BAB III

### ANALISA DAN PERANCANGAN SISTEM

#### 3.1 Langkah-langkah Penyelesaian Masalah

Agar pembuatan aplikasi sesuai dengan yang diinginkan, terencana, dan terdokumentasi dengan baik, maka dibuat langkah-langkah dalam penyelesaian permasalahan. Langkah-langkah tersebut, yaitu :

##### A. Analisa Kebutuhan Sistem

Pada tahapan ini dilakukan penelitian dan pengamatan tentang apa yang menjadi permasalahan dan kebutuhan dari penyandang tunanetra. Dari permasalahan dan kebutuhan tersebut dapat disimpulkan solusi dari permasalahan tersebut.

##### B. Gambaran Umum Sistem

Pada tahapan ini berisikan tentang gambaran umum dari sistem aplikasi yang akan dibuat. Gambaran tersebut dibuat berdasarkan atas permasalahan dan kebutuhan, serta dari solusi yang dihasilkan.

##### C. Rancangan Aplikasi

Pada tahap ini merupakan pengembangan dari gambaran umum sistem. Di tahap ini lebih dijelaskan lebih mendetil tentang isi dari sistem aplikasi yang dibuat. Agar lebih jelas maka pada tahap ini banyak mencantumkan diagram-diagram.

##### D. Desain *Input* dan *Output*

Pada tahap ini dibuat desain dari bentuk rancangan *input* dan *output* dari aplikasi yang dibuat.

#### E. Desain Uji Coba

Pada tahap desain uji coba, maka dibuat desain uji coba semua fungsi dari aplikasi akan dicoba untuk dijalankan. Untuk mengetahui apakah semua fungsi telah berjalan dengan baik.

### 3.2 Analisa Kebutuhan Sistem

Berdasarkan pengamatan yang dilakukan, kesimpulan yang dapat diambil adalah para penyandang tunanetra membutuhkan informasi waktu dan lokasi, secara mandiri. Hasil kesimpulan lainnya adalah bahwa hampir semua orang mempunyai perangkat bergerak, termasuk penyandang tunanetra.

Dengan berkembangnya teknologi informasi, perangkat bergerak seperti *smartphone* mengalami pertumbuhan yang sangat pesat. Perangkat bergerak sekarang mampu menjalankan banyak aplikasi dengan fitur yang canggih. Akan tetapi *smartphone* yang banyak beredar sekarang menggunakan layar sentuh yang tentunya sulit untuk digunakan bagi para penyandang tunanetra.

Dari pengamatan di atas, maka dapat disimpulkan bahwa para penyandang tunanetra membutuhkan *smartphone* yang dapat memberikan informasi waktu dan lokasi kepada mereka secara mandiri. *Smartphone* tersebut diharuskan untuk dapat berinteraksi dengan pengguna tunanetra dengan tanpa menggunakan interaksi *visual*, melainkan dengan interaksi sentuhan dan suara.

Untuk mengatasi masalah kebutuhan penyandang tunanetra akan informasi waktu dan lokasi secara mandiri, dan dengan menggunakan perkembangan *smartphone*. Maka peneliti mencoba mengembangkan aplikasi “Penunjuk Waktu dan Lokasi Untuk Penyandang Tunanetra Pada Perangkat Bergerak Berbasis

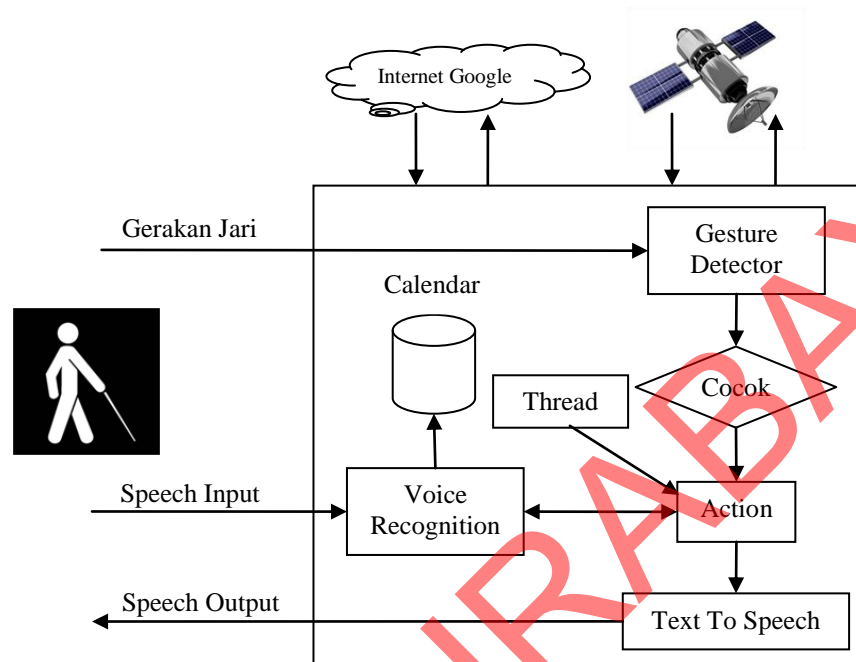
Android". Untuk mempermudah penggunaan aplikasi bagi tunanetra pada *smartphone* dengan layar sentuh, maka aplikasi yang dikembangkan ini menggunakan fitur *gesture detector*, yang berfungsi untuk mendeteksi gerakan jari dari pengguna pada layar sentuh dan juga berfungsi sebagai alat *input*. Selain itu aplikasi yang dikembangkan mempunyai kelebihan yaitu menggunakan interaksi suara sebagai *input* dan *output* aplikasi. Sehingga penyandang tunanetra yang memiliki kekurangan dalam hal penglihatan dapat terbantu dengan fitur-fitur tersebut.

Dalam pengimplemetasiannya, aplikasi ini digunakan pada *smartphone* Android. Versi Android yang digunakan adalah versi 2.3 (*gingerbread*). Dalam penggunaannya, aplikasi ini membutuhkan fitur *voice recognition* yang merupakan bawaan dari Google *search*, fitur tersebut digunakan sebagai alat interaksi *input* bagi pengguna tunanetra. *Text to speech* yang telah ada sejak Android versi 1.6 digunakan sebagai alat interaksi *output* suara bagi pengguna tunanetra, *GPS receiver* digunakan untuk mengambil nilai posisi (*altitude* dan *longitude*) perangkat bergerak, dan aplikasi ini membutuhkan koneksi internet. Koneksi internet dibutuhkan untuk mengaktifkan fitur *voice recognition* dan untuk melakukan sinkronisasi data GPS dengan *Google map*.

### 3.3 Gambaran Umum Sistem

Aplikasi *mobile* yang dikembangkan ini dapat membantu para penyandang tunanetra untuk dapat mengetahui informasi akan waktu dan lokasi secara mandiri. Aplikasi yang dibuat dilengkapi dengan fitur *gesture detector*, *voice*

*recognition* dan *text to speech*, ketiga fitur tersebut berfungsi sebagai alat *input* dan *output* bagi pengguna tunanetra.



Gambar 3.1 Gambaran Umum Sistem

Pada gambar 3.1 Proses dimulai ketika pengguna menyentuh layar dan menggerakkan jarinya, atau melakukan *double tap*. Pengguna tunanetra dapat mengetahui informasi waktu dan lokasi dengan cara menggunakan gerakan jari. Pendeteksian gerakan jari menggunakan fitur *gesture detector*, gerakan yang disimpan adalah gerakan ke atas, bawah, samping kanan, samping kiri, dan *double tap*. *Input* suara digunakan pada fungsi menambah alarm saja. Dari gerakan jari tersebut, aplikasi akan mendeteksinya dan kemudian aplikasi akan bereaksi sesuai dengan fungsi yang ada dalam aplikasi. Proses pendeteksian *input* suara pada fungsi menambah alarm menggunakan fitur *voice recognition*, fitur ini membutuhkan koneksi internet. Fitur ini digunakan untuk *input* waktu dan pesan pada jam alarm. Hasil *output* dari proses diatas berupa suara kepada pengguna

yang berisikan tentang informasi waktu atau lokasi. *Thread* pada gambar 3.1 diatas berfungsi sebagai *timer* yang akan memberikan informasi waktu jam tiap periode (15 menit).

### 3.3.1 Fungsi Aplikasi

Fungsi aplikasi yang dibuat ini adalah memberikan informasi kepada pengguna tunanetra mengenai informasi waktu dan lokasi dimana ia berada berdasarkan data yang didapat dari GPS.

Informasi waktu berisikan hari, tanggal, dan jam didapatkan dari waktu sistem. Sedangkan informasi lokasi berisikan nama jalan/alamat yang didapatkan dari *map server* Google dengan bantuan sinkronisasi dari GPS.

Informasi waktu didapatkan dengan cara menggerakkan jari. Menggerakkan jari ke kiri berarti tanggal, menggerakkan jari ke atas berarti jam. Informasi waktu jam juga akan diinformasikan tiap periode. Informasi lokasi didapatkan dengan cara menggerakkan jari ke kanan. Ke bawah untuk bantuan. Yang terakhir adalah *double tap* untuk menambah alarm.

### 3.3.2 Memasukkan Suara

Pada aplikasi ini setelah melakukan *double tap*, tampilan *voice recognizer intent* akan muncul dan pengguna dapat memasukkan *input* suara yang kemudian akan dideteksi oleh *voice recognition*. Suara yang telah dideteksi kemudian akan diubah menjadi teks. Teks yang dihasilkan akan digunakan sebagai *input* waktu dan pesan pada alarm.

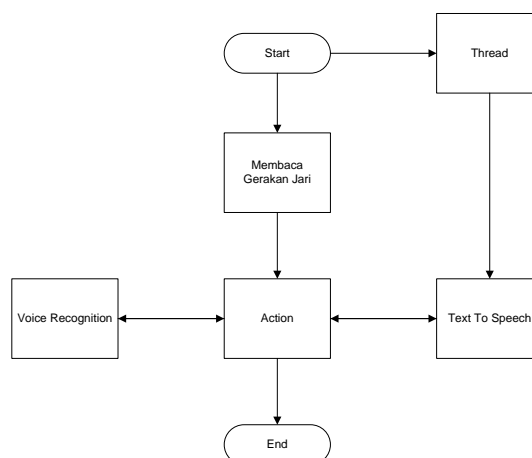
### 3.4 Rancangan Aplikasi

Rancangan aplikasi ini akan memberikan gambaran dari aplikasi yang dibuat dengan menggunakan desain sistem yang berorientasi objek, yaitu UML, serta digambarkan menggunakan Star UML. Adapun desain yang dibuat antara lain:

- a. *Flowchart*
- b. *Use Case Diagram*
- c. *Flow of Event*
- d. *Activity Diagram*
- e. *Sequence Diagram*
- f. *Class Diagram*
- g. CDM dan PDM

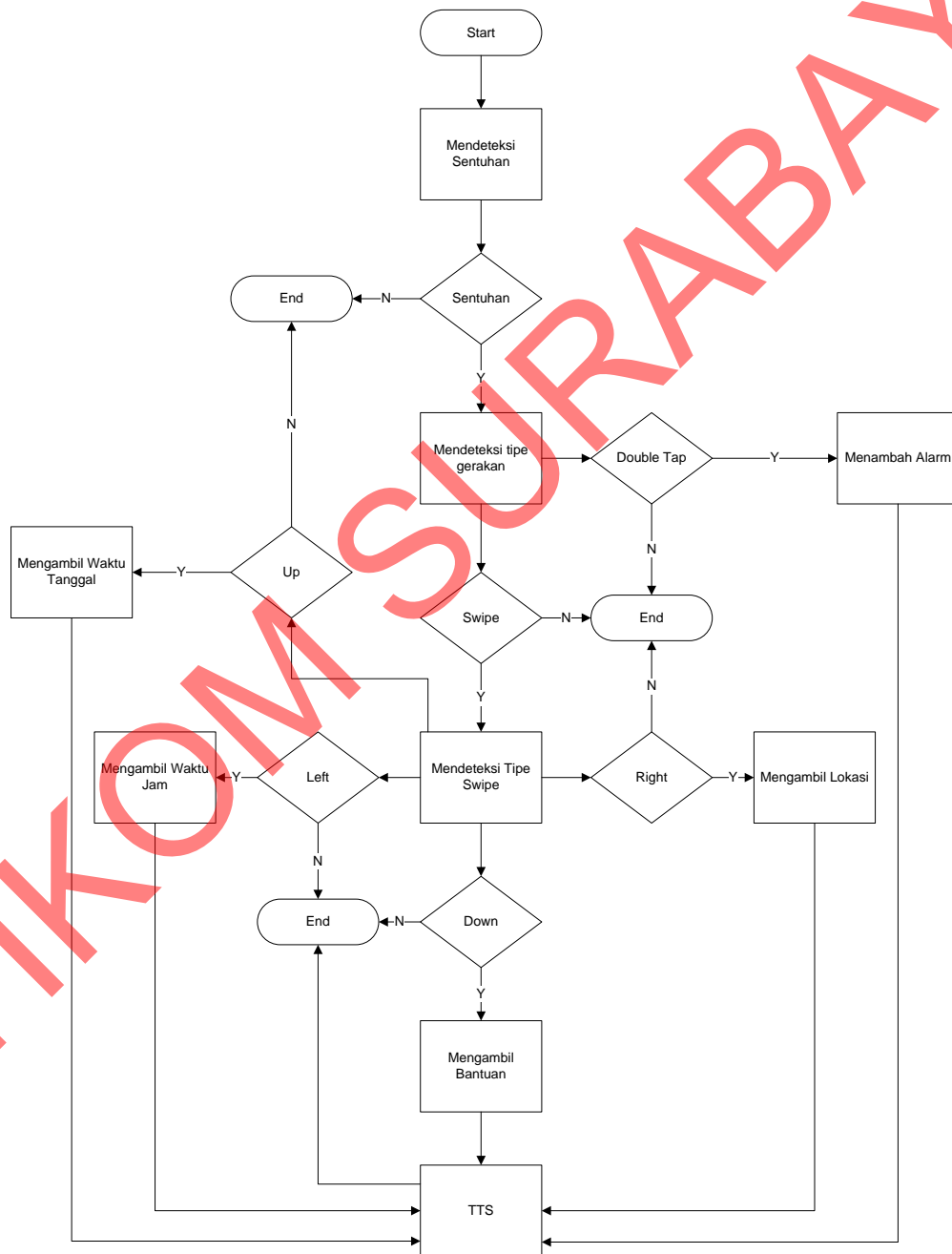
#### 3.4.1 Flowchart

Agar dapat lebih memahami proses apa saja yang berjalan pada aplikasi maka dibuatlah *flowchart* yang berisikan detail tentang proses-proses apa saja yang ada dan dijalankan pada aplikasi.



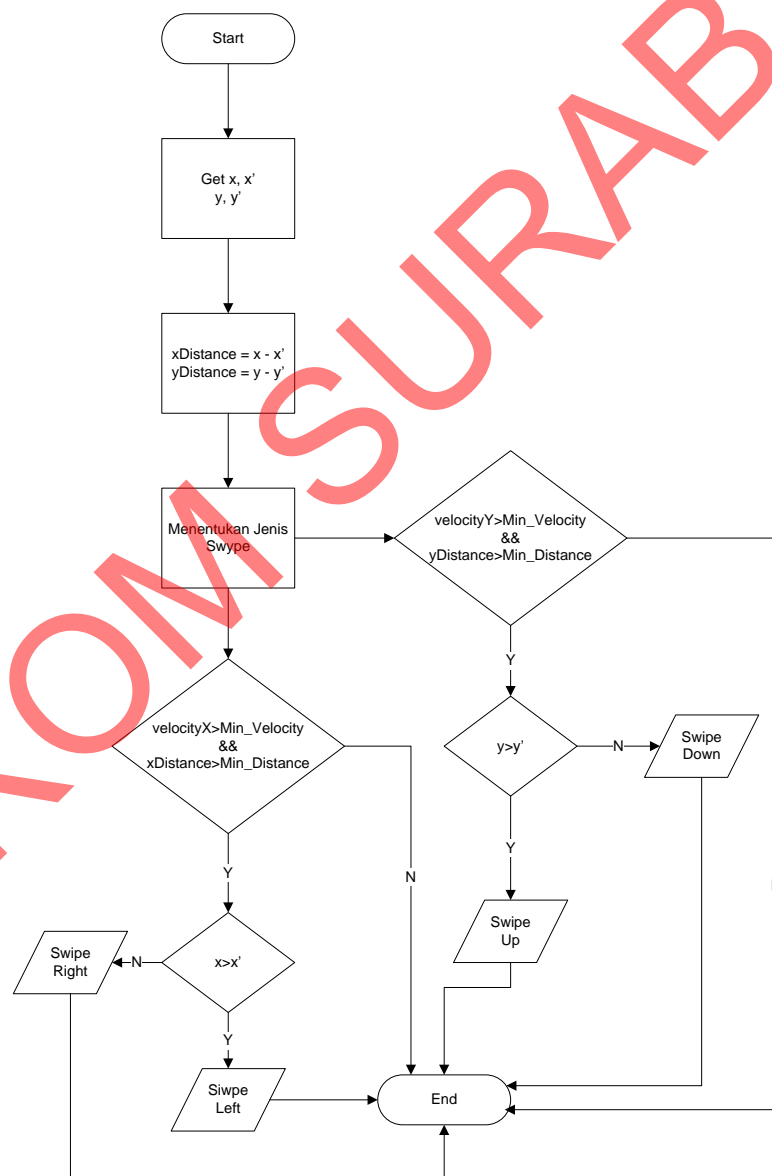
Gambar 3.2 Flowchart Utama Aplikasi

Pada gambar 3.2 dapat dilihat *flowchart* utama dari aplikasi. Dapat dilihat aplikasi memiliki beberapa proses utama yaitu mendeteksi gerakan jari, *action*, *voice recognition*, *text to speech*, dan *thread*. Pada proses *action* terdapat 5 fungsi utama aplikasi, yaitu penunjuk waktu tanggal, penunjuk waktu jam, penunjuk lokasi, menambah alarm, dan bantuan.



Gambar 3.3 Flowchart Mendeteksi Gerakan Jari

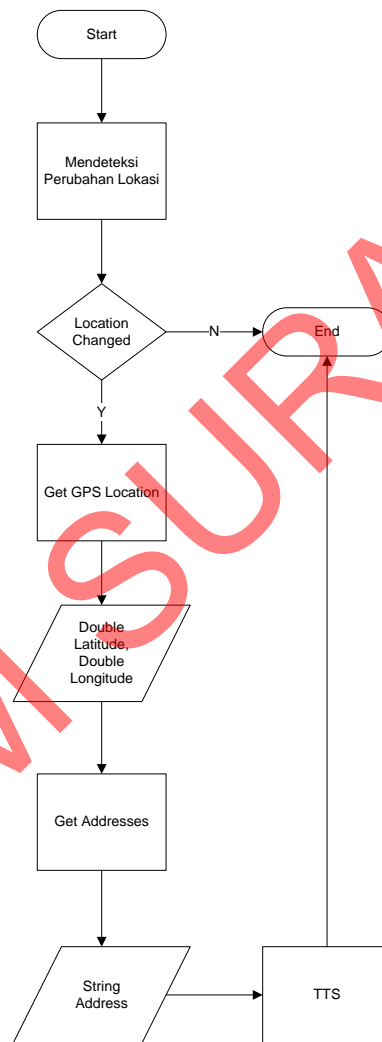
Pada gambar sebelumnya dapat dilihat *flowchart* dari proses deteksi gerakan yang dilakukan pada layar sentuh. Dimulai dengan mendeteksi sentuhan pada layar, kemudian dilanjutkan dengan mendeteksi tipe gerakan. Ada dua tipe gerakan, yaitu gerakan *double tap* atau *swipe*. Gerakan *swipe* dibagi lagi menjadi 4, yaitu atas, bawah, kanan, dan kiri. Berdasarkan gerakan-gerakan tersebut fungsi-fungsi dari aplikasi akan dijalankan.



Gambar 3.4 Flowchart Mendeteksi Tipe Swipe



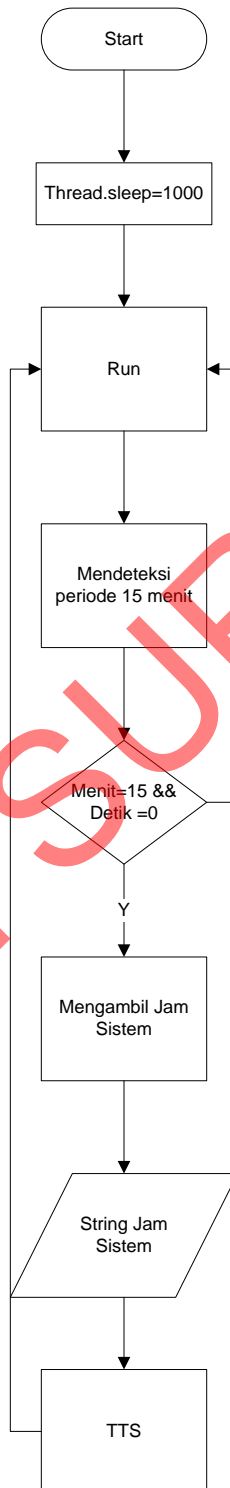
Pada gambar 3.4 dijelaskan bagaimana proses deteksi tipe gerakan *swipe*. Proses deteksi dilakukan dengan mengambil posisi awal sentuhan jari pada layar disimbolkan dengan  $x,y$ . Kemudian mengambil posisi akhir sentuhan jari disimbolkan dengan  $x',y'$ . Setelah didapatkan keempat nilai tersebut, maka dilakukan perbandingan untuk menentukan arah gerakan jari pengguna.



Gambar 3.5 Flowchart Penunjuk Lokasi

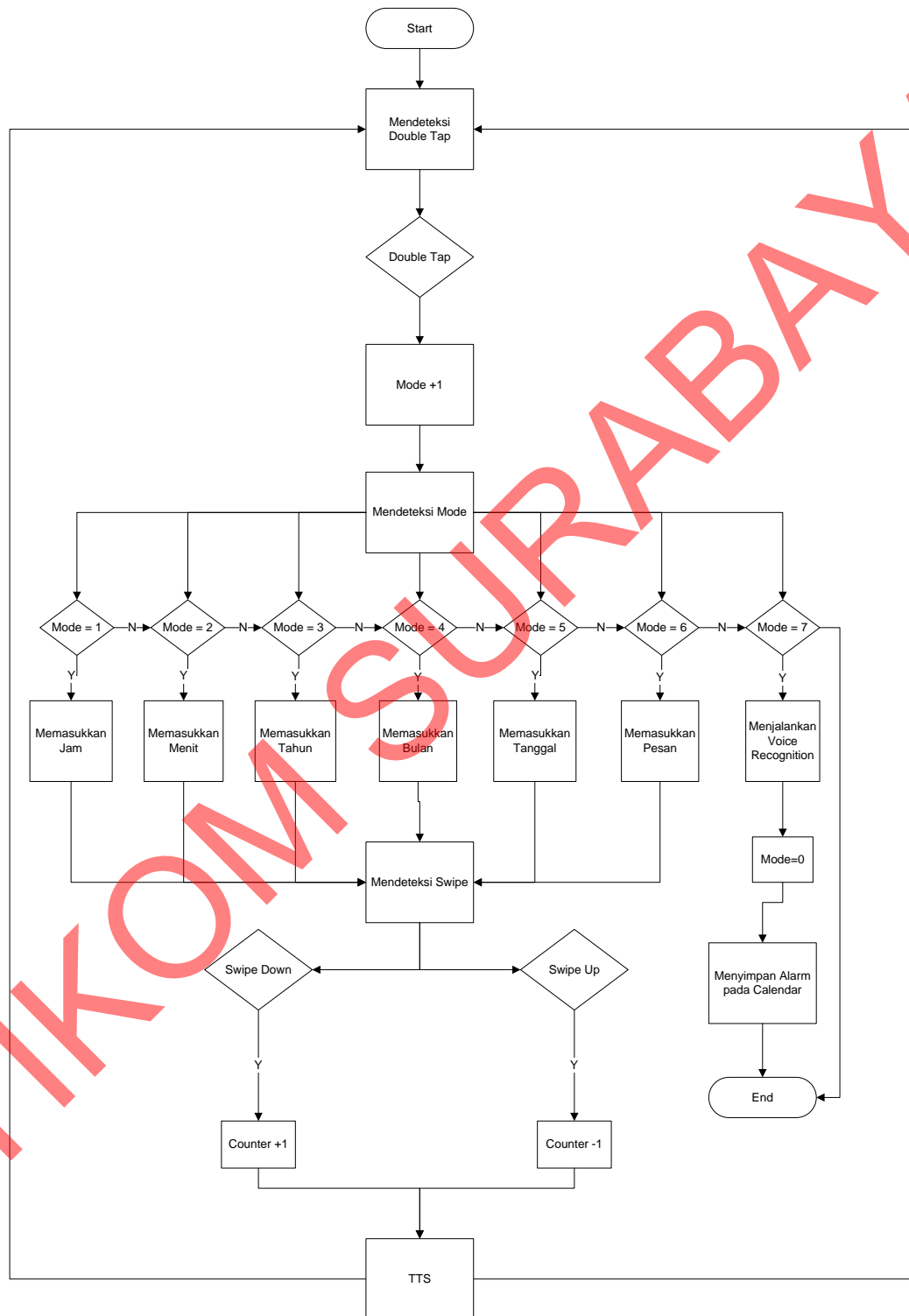
Pada gambar diatas dapat dilihat proses dari penunjuk lokasi. Dimulai dengan pendeteksian perubahan lokasi. Jika ada perubahan, maka dilakukan pengambilan nilai *altitude* dan *longitude*, dengan menggunakan *GPS receiver*.

Setelah kedua nilai tersebut didapatkan. Maka dilakukan *synchronize* dengan Google *Map* untuk mendapatkan alamat dari lokasi tersebut.



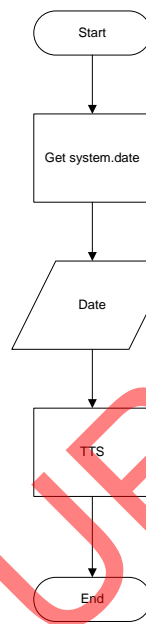
Gambar 3.6 Flowchart Thread

Gambar diatas merupakan proses dari *thread* yang berfungsi untuk menginformasikan waktu jam tiap 15 menit.

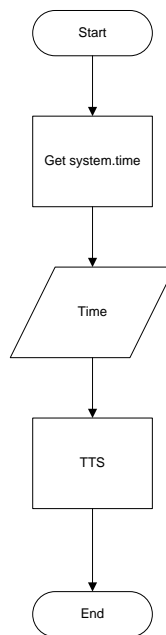


Gambar 3.7 Flowchart Menambah Alarm

Gambar diatas merupakan *flowchart* dari proses menambah alarm. Proses dilakukan secara berurutan mulai dari *mode 1* sampai *mode 7*. *Counter* berfungsi untuk menaikkan nilai dan menurunkan nilai dari angka yang akan dimasukkan ke dalam alarm.



Gambar 3.8 Flowchart Mengambil Waktu Tanggal

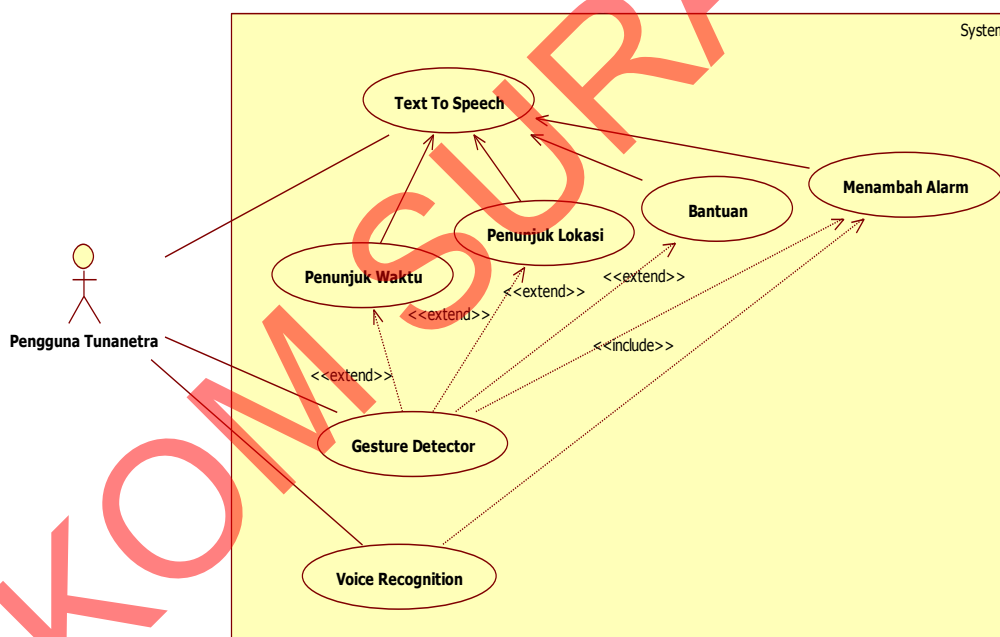


Gambar 3.9 Flowchart Mengambil Waktu Jam

### 3.4.2 Use Case Diagram

Kemampuan aplikasi untuk dapat berinteraksi dengan pengguna digambarkan dalam *use case diagram* pada gambar berikut. Terdapat 1 aktor pada *use case* tersebut, dan terdapat 5 buah *use case*, yaitu:

1. *Voice Recognition*
2. Penunjuk Waktu
3. Penunjuk Lokasi
4. Menambah Alarm
5. Bantuan



Gambar 3.10 Use Case Aplikasi Penunjuk Waktu dan Lokasi

Pada gambar di atas dapat dijelaskan bahwa pengguna tunanetra dapat menggunakan aplikasi dengan cara memasukkan gerakan jari melalui layar sentuh. Gerakan tersebut akan dideteksi oleh *use case gesture detector*. *Use case* penunjuk waktu, penunjuk lokasi, menambah alarm, dan bantuan merupakan

perluasan dari *use case gesture detector*. Dan *use case voice recognition* termasuk di dalam *use case* menambah alarm, karena otomatis dijalankan *use case* menambah alarm dijalankan dan digunakan sebagai alat *input* pada *use case* menambah alarm.

### 3.4.3 Flow of Event

Dari *use case* yang ada, dibutuhkan *flow of event* untuk menjelaskan spesifikasi proses/aliran yang terjadi pada tiap *use case*, serta untuk mendokumentasikan aliran logika dalam *use case*. *Flow of event* meliputi deskripsi singkat, kondisi awal, aliran kejadian utama, aliran kejadian alternatif, kondisi akhir. *Flow of event* yang dibuat adalah *flow of event* untuk *use case* penunjuk waktu, penunjuk lokasi, menambah alarm, dan bantuan.

#### A. Flow of Event untuk Use Case Penunjuk Waktu

*Flow of event* untuk *use case* penunjuk waktu tanggal dapat dilihat pada tabel 3.1.

Tabel 3.1 Flow of Event Penunjuk Waktu Tanggal

<b>Deskripsi</b>	<i>Use case</i> penunjuk waktu memungkinkan pengguna untuk dapat mengetahui informasi waktu tanggal.	
<b>Kondisi Awal</b>	-	
<b>Kondisi Akhir</b>	Aplikasi berhasil memberi informasi waktu tanggal melalui media suara	
<b>Aliran Kejadian Utama</b>		<b>Aksi Pemakai</b>
	<b>1</b>	<i>Use case</i> dimulai ketika pengguna menggerakkan jari ke kiri
		<b>Respon Sistem</b>
		Sistem memberikan informasi tanggal

*Flow of event* untuk *use case* penunjuk waktu jam dapat dilihat pada tabel

3.2.

Tabel 3.2 Flow of Event Penunjuk Waktu Jam

<b>Deskripsi</b>	<i>Use case</i> penunjuk waktu memungkinkan pengguna untuk dapat mengetahui informasi waktu jam.		
<b>Kondisi Awal</b>	-		
<b>Kondisi Akhir</b>	Aplikasi berhasil memberi informasi waktu jam melalui media suara		
<b>Aliran Kejadian Utama</b>		<b>Aksi Pemakai</b>	<b>Respon Sistem</b>
	1	<i>Use case</i> dimulai ketika pengguna menggerakkan jari ke atas	Sistem memberikan informasi jam
	2	-	Sistem memberikan informasi jam tiap periode 15 menit

#### B. Flow of Event untuk Use Case Penunjuk Lokasi

*Flow of event* untuk *use case* penunjuk lokasi dapat dilihat pada tabel 3.3.

Tabel 3.3 Flow of Event Penunjuk Lokasi

<b>Deskripsi</b>	<i>Use case</i> penunjuk lokasi memungkinkan pengguna untuk dapat mengetahui informasi lokasi.		
<b>Kondisi Awal</b>	-		
<b>Kondisi Akhir</b>	Aplikasi berhasil memberi informasi lokasi melalui media suara		
<b>Aliran Kejadian Utama</b>		<b>Aksi Pemakai</b>	<b>Respon Sistem</b>
	1	<i>Use case</i> dimulai ketika pengguna menggerakkan jari ke arah kanan	Sistem memberikan informasi lokasi

#### C. Flow of Event untuk Use Case Menambah Alarm

*Flow of event* untuk *use case* menambah alarm dapat dilihat pada tabel 3.4.

Tabel 3.4 Flow of Event Menambah Alarm

<b>Deskripsi</b>	<i>Use case</i> menambah alarm memungkinkan pengguna untuk dapat menambah alarm	
<b>Kondisi Awal</b>	-	
<b>Kondisi Akhir</b>	Aplikasi berhasil memasukkan waktu dan pesan alarm melalui media suara	
<b>Aliran Kejadian Utama</b>		<b>Aksi Pemakai</b>
	<b>1</b>	<i>Use case</i> dimulai ketika pengguna melakukan <i>double tap</i>
		<b>Respon Sistem</b>
		Sistem menjalankan proses input dan fitur <i>voice recognition</i> , sehingga pengguna dapat memasukkan waktu dan pesan alarm ke sistem

#### D. Flow of Event untuk Use Case Bantuan

*Flow of event* untuk *use case* bantuan dapat dilihat pada tabel 3.5.

Tabel 3.5 Flow of Event Bantuan

<b>Deskripsi</b>	<i>Use case</i> bantuan memberikan bantuan kepada pengguna melalui media suara	
<b>Kondisi Awal</b>	-	
<b>Kondisi Akhir</b>	Aplikasi berhasil memberi informasi bantuan melalui media suara	
<b>Aliran Kejadian Utama</b>		<b>Aksi Pemakai</b>
	<b>1</b>	<i>Use case</i> dimulai ketika pengguna menggerakkan jari ke arah bawah
		<b>Respon Sistem</b>
		Sistem memberikan informasi bantuan

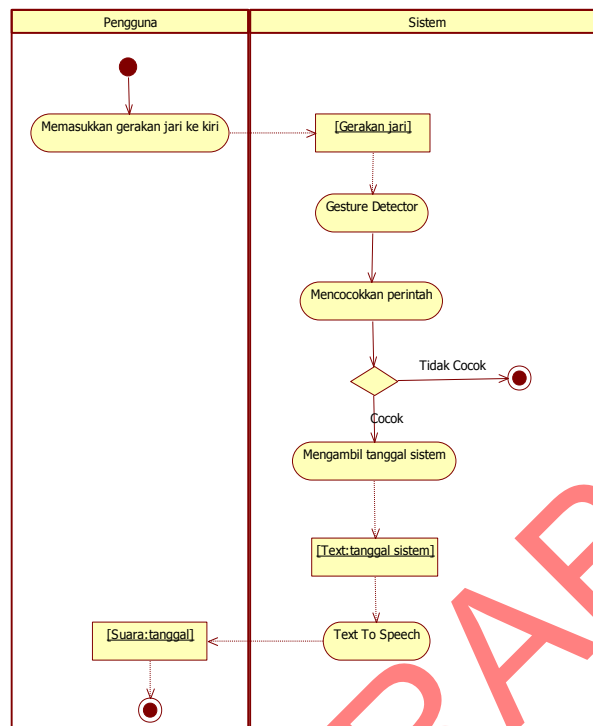
#### 3.4.4 Activity Diagram

Dari *use case* yang ada dibutuhkan *activity* diagram untuk menjelaskan proses/aliran proses yang terjadi pada tiap *use case*.

##### A. Activity Diagram untuk Use Case Penunjuk Waktu

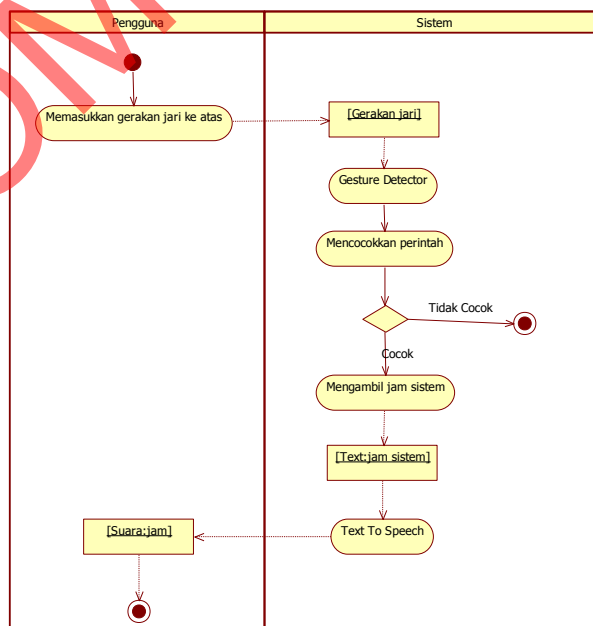
Proses pada gambar 3.11 dimulai dengan pengguna ingin mendapatkan informasi waktu tanggal dengan cara menggerakkan jari ke kiri. Gerakan jari akan dideteksi oleh *gesture detector*.



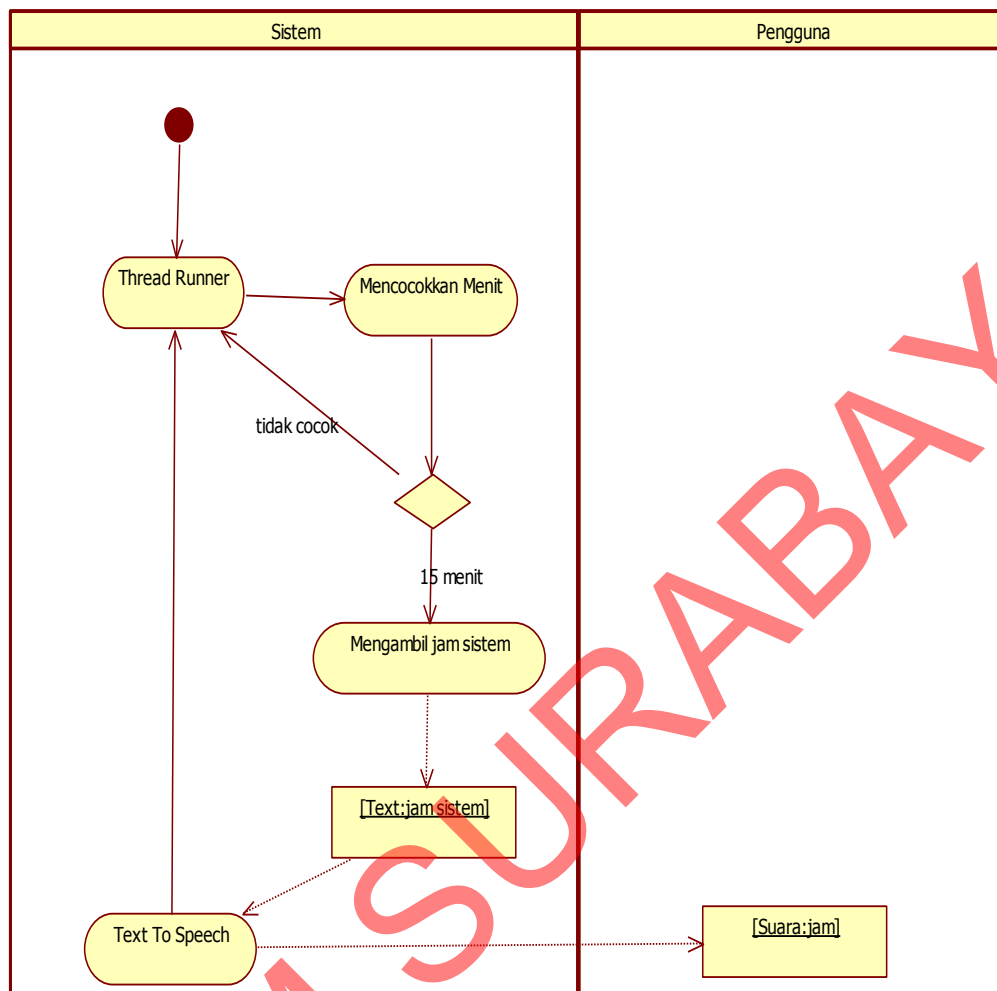


Gambar 3.11 Activity Diagram Penunjuk Waktu Tanggal

Sedangkan untuk mendapatkan informasi waktu jam pengguna dapat dilakukan dengan cara menggerakkan jari ke atas.



Gambar 3.12 Activity Diagram Penunjuk Waktu Jam



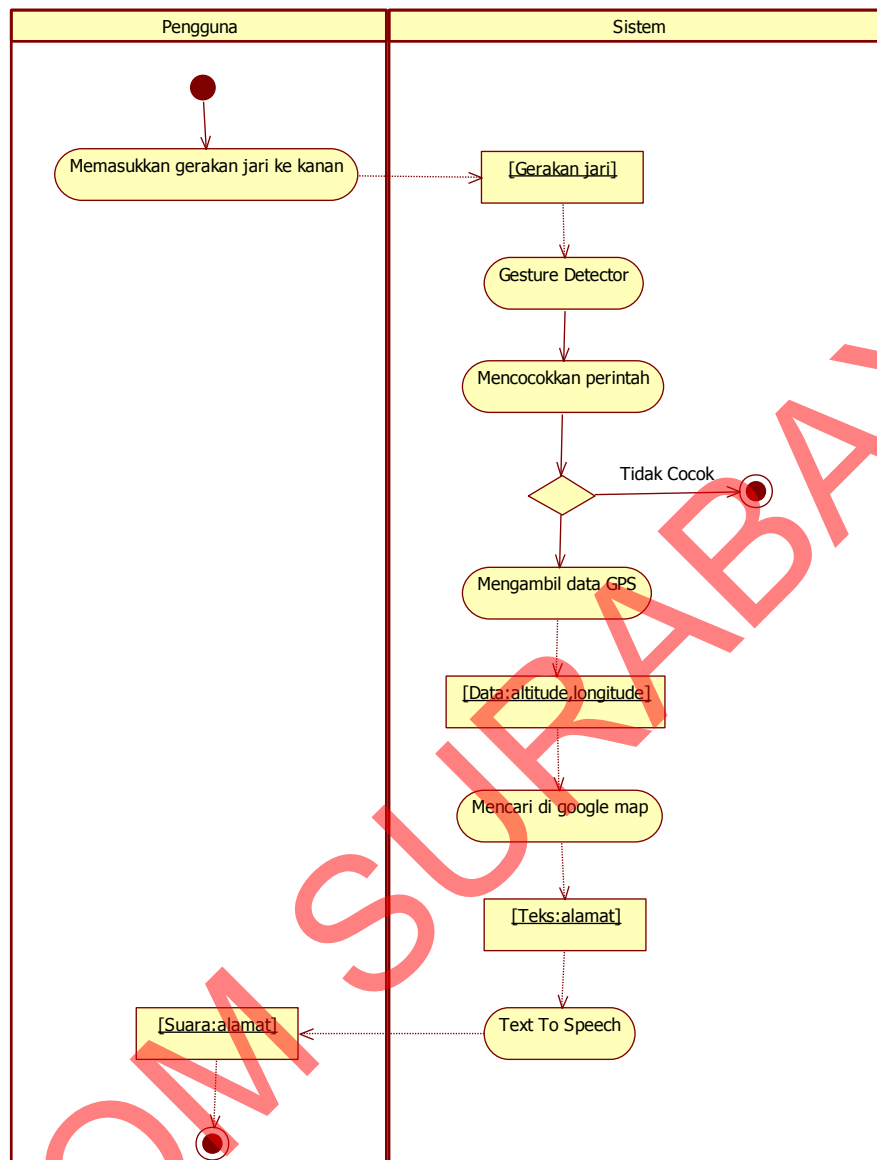
Gambar 3.13 Activity Diagram Peningat Waktu Jam (Thread)

### B. Activity Diagram untuk Use Case Penunjuk Lokasi

Proses pada gambar 3.14 dimulai ketika pengguna ingin mendapatkan informasi akan lokasi dimana ia berada dengan menggerakkan jari ke arah kanan.

*Gesture detector* akan mencocokkan gerakan jari tersebut dengan fungsi yang ada.

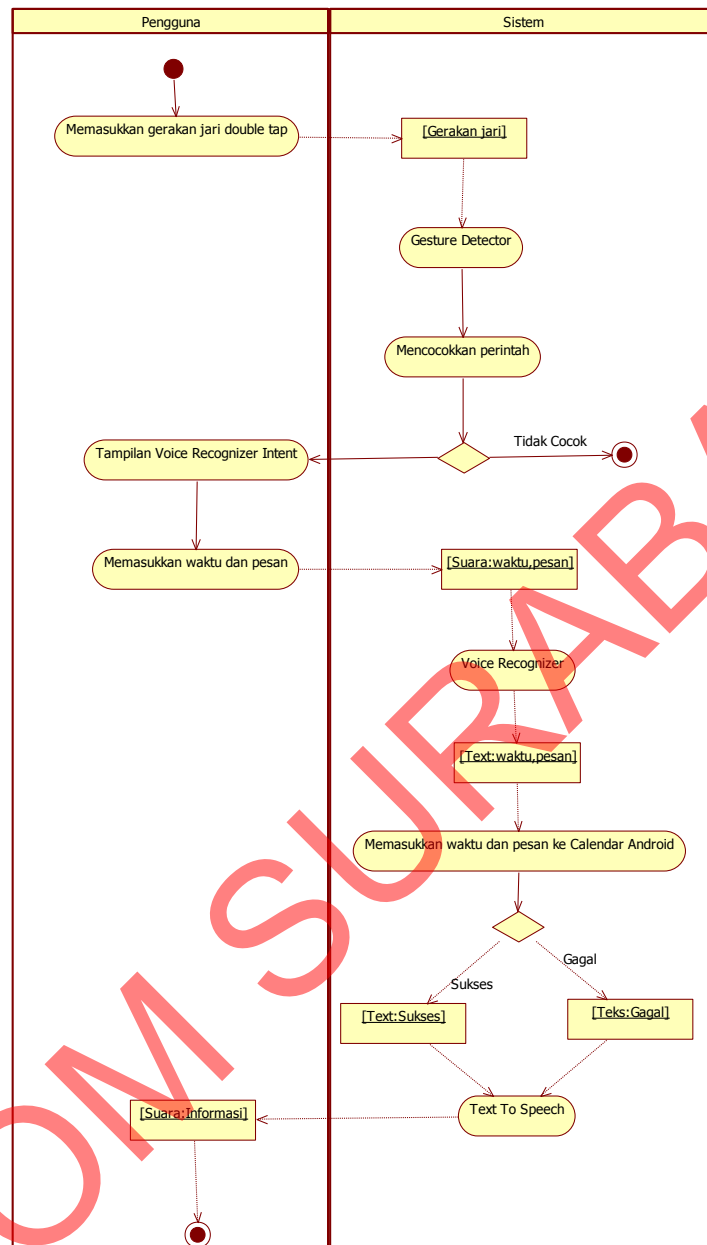
Sistem akan mengambil data alamat melalui *Google map*, dengan bantuan data dari GPS.



Gambar 3.14 Activity Diagram Penunjuk Lokasi

### C. Activity Diagram untuk Use Case Menambah Alarm

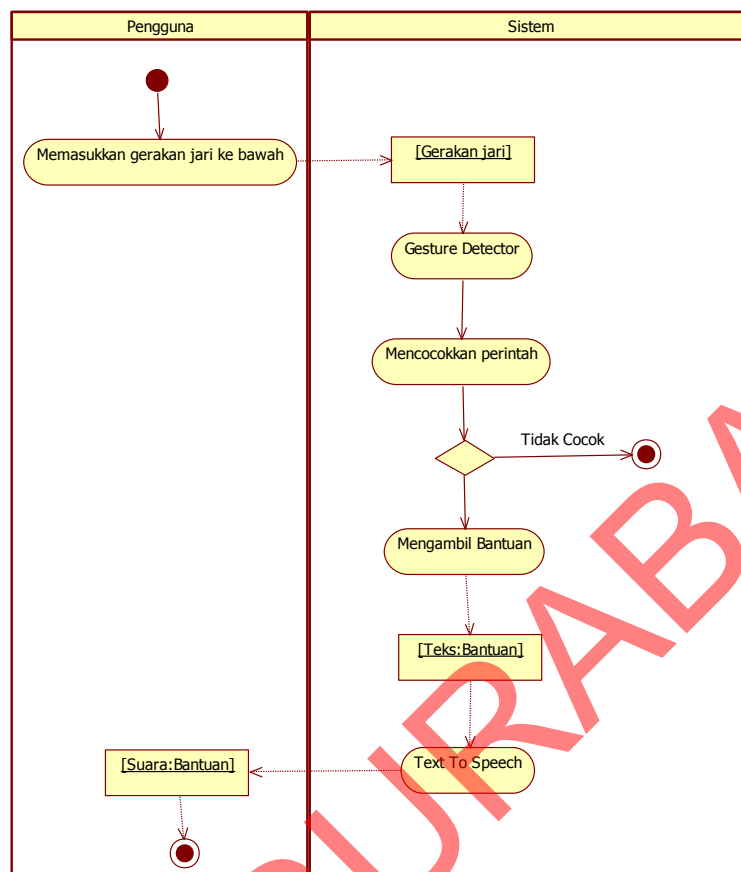
Proses pada gambar 3.15 dimulai dengan pengguna melakukan *double tap* pada layar. *Voice recognition* akan muncul dan pengguna diminta memasukkan waktu dan pesan alarm.



Gambar 3.15 Activity Diagram Menambah Alarm

#### D. Activity Diagram untuk Use Case Bantuan

Proses pada gambar 3.16 dimulai ketika pengguna menginginkan bantuan dari sistem, dengan cara menggerakkan jari ke bawah. Sistem akan memberikan bantuan dalam bentuk suara kepada pengguna.

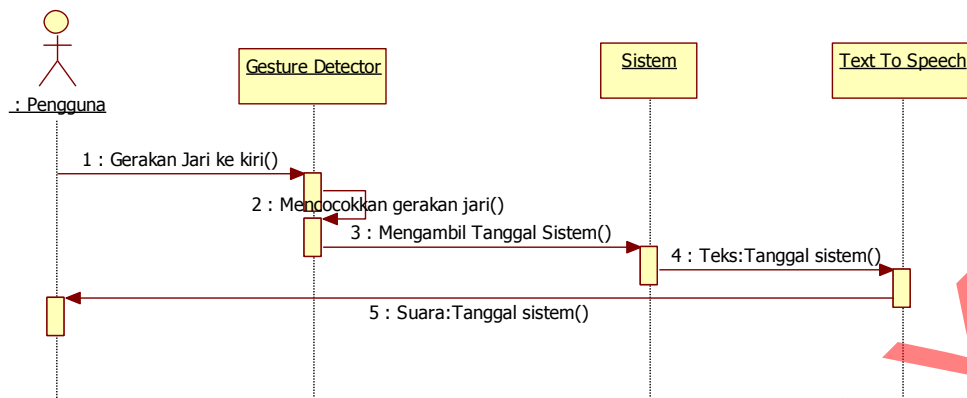


Gambar 3.16 Activity Diagram Bantuan

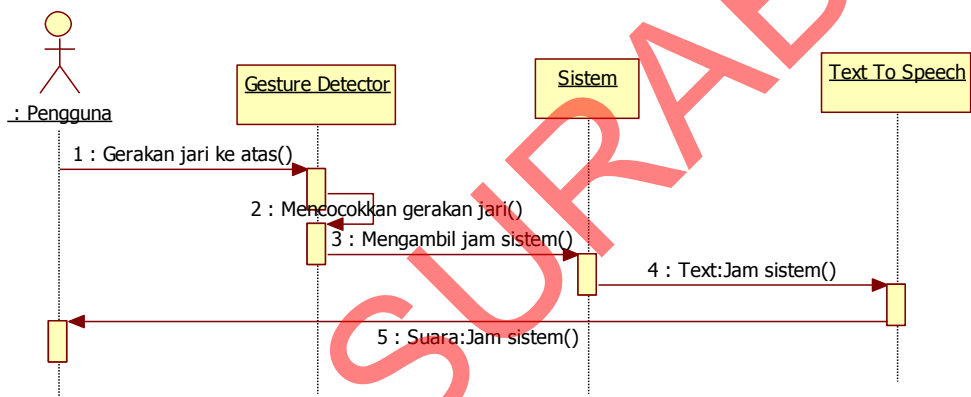
### 3.4.5 Sequence Diagram

#### A. Sequence Diagram untuk Use Case Penunjuk Waktu

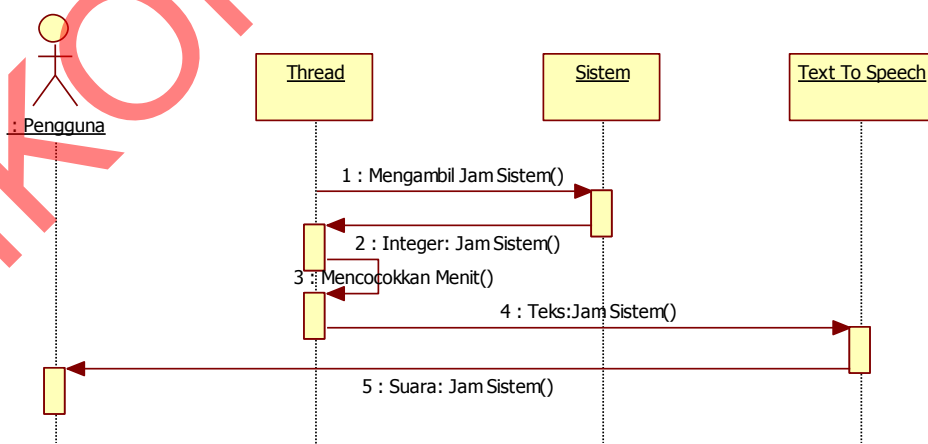
Pada gambar 3.17 dan 3.18 dimulai dengan pengguna menggerakkan jari ke kiri untuk tanggal dan ke kanan untuk jam. Proses berikutnya *gesture* detector yang akan mendeteksi gerakan jari tersebut. Kemudian sistem akan mengambil waktu tanggal atau jam pada sistem dan merubahnya menjadi bentuk suara dengan fitur *text to speech*. Sedangkan untuk pengingat waktu pada gambar 3.19 akan otomatis berjalan setiap periode 15 menit.



Gambar 3.17 Sequence Diagram Penunjuk Waktu Tanggal



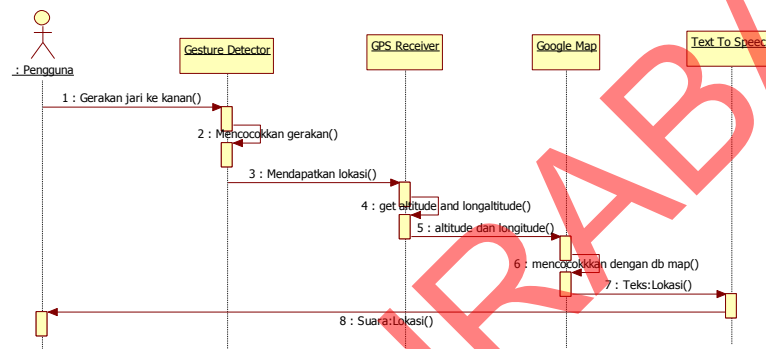
Gambar 3.18 Sequence Diagram Penunjuk Waktu Jam



Gambar 3.19 Sequence Diagram Peningat Waktu Jam (Thread)

## B. Sequence Diagram untuk Use Case Penunjuk Lokasi

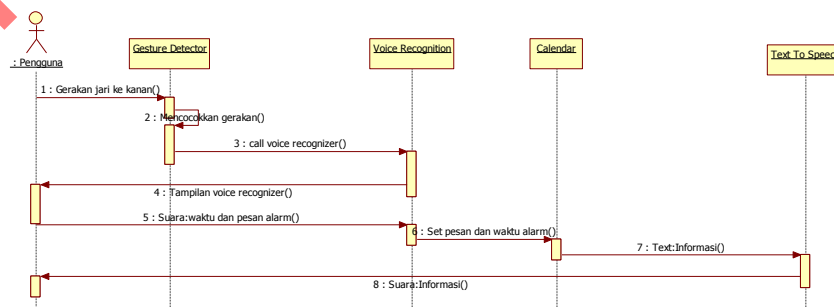
Pada gambar 3.20 dimulai ketika pengguna menggerakkan jari ke kanan pada layar. Gerakan yang dideteksi oleh *gesture detector*, akan dicocokkan. Jika sesuai sistem akan mengambil data GPS dan mencocokkannya dengan data yang ada pada *Google map*. Data teks alamat yang didapatkan kemudian dirubah menjadi bentuk suara oleh *text to speech*



Gambar 3.20 Sequence Diagram Penunjuk Lokasi

## C. Sequence Diagram untuk Use Case Menambah Alarm

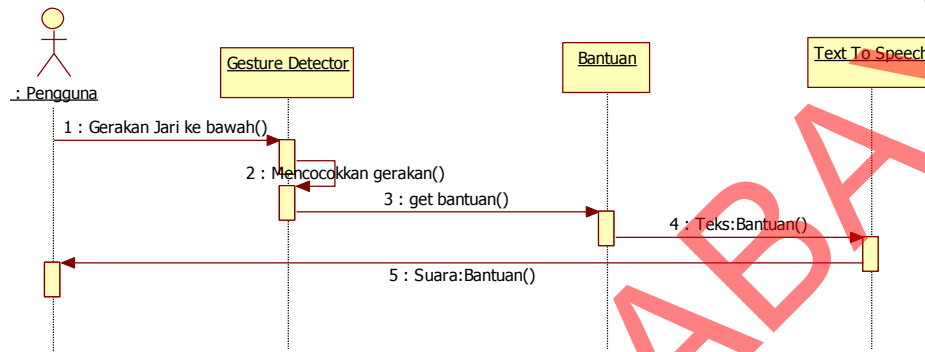
Pada gambar 3.21 dimulai dengan pengguna melakukan double tap pada layar. *Voice recognizer* yang dipanggil oleh *voice recognition* menampung dan menerjemahkan waktu dan pesan alarm. Maka proses selanjutnya adalah mengirimkan data waktu dan pesan alarm ke *Android calendar*.



Gambar 3.21 Sequence Diagram Menambah Alarm

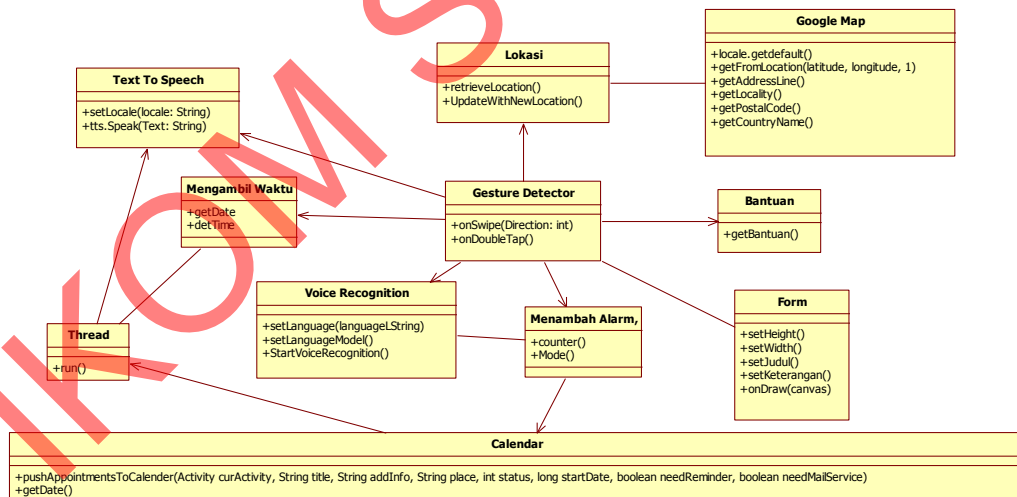
#### D. Sequence Diagram untuk Use Case Bantuan

Pada gambar 3.22 dimulai ketika pengguna menggerakkan jari ke bawah ketika menyentuh layar, kemudian sistem akan memberikan bantuan dalam bentuk suara hasil terjemahan dari *text to speech*.



Gambar 3.22 Sequence Diagram Bantuan

#### 3.4.6 Class Diagram



Gambar 3.23 Class Diagram

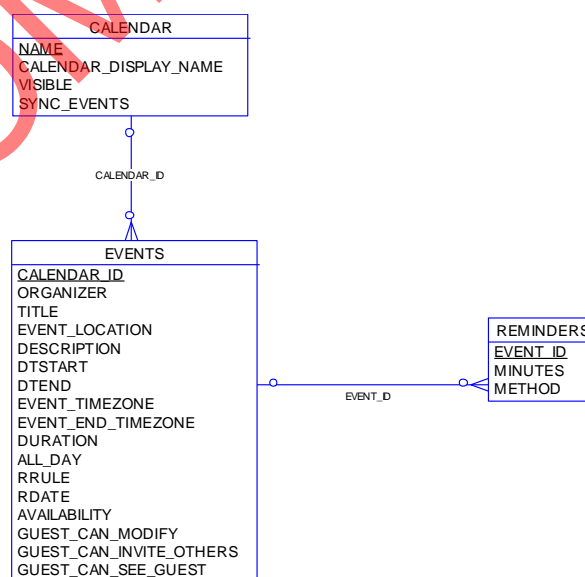
*Class Voice Recognition* digunakan untuk memproses suara pengguna menjadi bentuk teks, sehingga teks yang berisikan waktu dan pesan alarm tersebut digunakan untuk *input* pada Android *calendar*. *Class Mengambil Waktu*



digunakan untuk mengambil waktu tanggal atau jam pada sistem. *Class* Lokasi berfungsi untuk mengambil data *altitude* dan *longitude* pada *GPS receiver* dan mengirimkannya ke *Google map*, *class Google map* kemudian mencocokkannya dan mendapatkan nilai alamatnya. *Class* Bantuan berfungsi untuk memberikan bantuan kepada pengguna. *Class* Form berfungsi sebagai pengatur tampilan layar dan juga berguna sebagai penyedia tempat dan nilai posisi gerakan jari untuk *class gesture detector*. *Class gesture detector* berfungsi untuk mendeteksi gerakan jari pengguna pada layar, kemudian berdasarkan gerakan tersebut fungsi aplikasi akan dijalankan. *Class text to speech* berfungsi sebagai penerjemah dari bentuk teks hasil dari *class* lain menjadi bentuk suara. *Class thread* digunakan sebagai *timer* yang berfungsi untuk menjalankan fungsi penunjuk waktu jam dengan periode 15 menit.

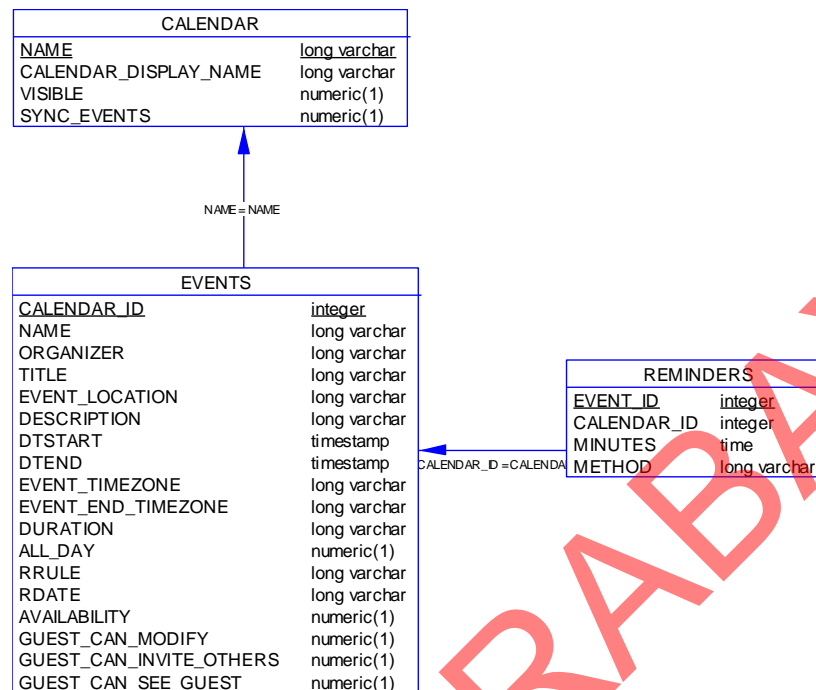
### 3.4.7 Entity Relationship Diagram

#### A. Conceptual Data Model (CDM)



Gambar 3.24 Conceptual Data Model (CDM)

## B. Physical Data Model (PDM)



Gambar 3.25 Physical Data Model (PDM)

### 3.5 Desain Input Output

Setelah melakukan perancangan aplikasi, tahap selanjutnya adalah membuat desain *input/output*. Perangkat lunak yang digunakan untuk membuat desain *input output* untuk tampilan adalah Microsoft *Visio 2007*.

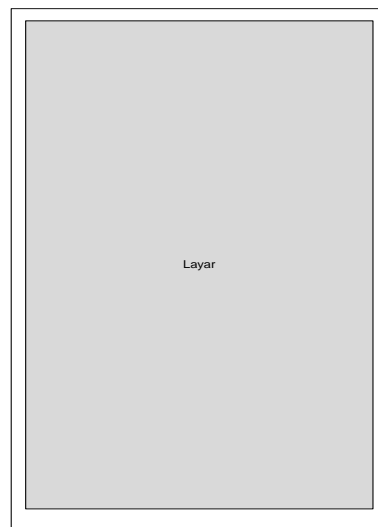
Dikarenakan pengguna adalah para penyandang tunanetra, maka penekanan desain *input* dan *output* pada aplikasi ini lebih kepada tiga fitur yang digunakan untuk berinteraksi dengan pengguna tunanetra. Fitur-fitur tersebut adalah *gesture detector*, *text to speech*, dan *voice recognition*.

Gambar 3.26 dibawah merupakan tampilan aplikasi ketika dijalankan. Dan merupakan satu-satunya tampilan yang ada. Untuk menjalankan fungsi dari aplikasi yang ada. Pengguna cukup menyentuh layar dan menggerakkan jari sambil menyentuh layar kearah kiri untuk mengetahui waktu tanggal, arah atas

untuk mengetahui waktu jam, ke arah kanan untuk mengetahui lokasi, ke arah bawah untuk mengaktifkan bantuan, dan *double tap* untuk menambah alarm.

Semua informasi akan dibacakan dengan menggunakan fitur *text to speech*. Khusus untuk fungsi menambah alarm, dimana terdapat *input* untuk pesan alarm. Suara pengguna akan dirubah menjadi bentuk teks menggunakan *voice recognition*. Kemudian teks tersebut digunakan sebagai pesan dalam alarm.

Suara informasi dibacakan oleh *text to speech* pada aplikasi untuk informasi waktu tanggal, contoh “17 Agustus 2012”, maka akan dibacakan “*seventeen august two thousand twelve*”. Untuk informasi waktu jam, contoh “12.15”, maka akan dibacakan “*twelve fifteen*”. Untuk informasi lokasi akan ditambahkan kalimat “*your current position is*” sebelum teks lokasi yang didapatkan. Untuk menambah alarm setelah melakukan *double tap*. Pengguna diminta untuk memasukkan jam, menit, tahun, bulan, tanggal, dan pesan alarm. Selain pesan alarm, proses *input* menggunakan bantuan fitur *gesture detector*, dengan cara menggerakkan jari kebawah untuk menaikkan nilai dan keatas untuk menurunkan nilai.



Gambar 3.26 Desain Halaman Utama

### 3.6 Desain Uji Coba

Pengujian pada desain uji coba ini dilakukan dengan tujuan untuk dapat mengetahui apakah fungsi-fungsi yang telah dimiliki oleh aplikasi penunjuk waktu dan lokasi untuk penyandang tunanetra pada perangkat bergerak berbasis Android ini telah dapat berjalan sesuai dengan apa yang diharapkan. Adapaun berbagai desain uji coba dari fungsi-fungsi yang diujikan antara lain :

#### A. Desain Uji Coba Fungsi Menjalankan Aplikasi Setelah Booting Up

Tabel 3.6 Desain Uji Coba Fungsi Aplikasi Setelah Booting Up

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
1	Menjalankan aplikasi setelah <i>booting up</i>	Menyalakan <i>smartphone</i>	Aplikasi berjalan setelah <i>smartphone</i> selesai <i>booting up</i>	

#### B. Desain Uji Coba Fungsi Penunjuk Waktu

Tabel 3.7 Desain Uji Coba Fungsi Penunjuk Waktu

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
2	Meminta informasi waktu "jam"	Menggerakkan jari ke kiri pada layar	Muncul suara berisikan informasi mengenai waktu jam	
3	Meminta informasi waktu "jam"	-	Muncul suara berisikan informasi mengenai waktu jam dengan periode 15 menit	
4	Meminta informasi waktu "tanggal"	Menggerakkan jari ke kanan pada layar	Muncul suara berisikan informasi waktu tanggal	

## C. Desain Uji Coba Fungsi Penunjuk Lokasi

Tabel 3.8 Desain Uji Coba Fungsi Penunjuk Lokasi

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
5	Meminta informasi lokasi	Menggerakkan jari ke kanan pada layar	Muncul suara berisikan informasi lokasi	

## D. Desain Uji Coba Fungsi Menambah Alarm

Tabel 3.9 Desain Uji Coba Fungsi Menambah Alarm

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
6	Menambahkan alarm	<i>Double tap</i> pada layar dan pengguna dapat memasukkan pesan suara	<i>Voice Recognizer</i> tampil. Alarm pada Android <i>Calendar</i> bertambah	

## E. Desain Uji Coba Fungsi Bantuan

Tabel 3.10 Desain Uji Coba Fungsi Bantuan

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
7	Meminta Bantuan	Menggerakkan jari ke bawah pada layar	Suara informasi bantuan berisikan informasi fungsi aplikasi	

F. Desain Uji Coba Tingkat Cepat Tangkap GPS

Tabel 3.11 Desain Uji Coba Tingkat Cepat Tangkap GPS

Test Case ID	Tempat Uji Coba	Waktu Tangkap
8	Lapangan tanpa halangan	
9	Lapangan dengan halangan gedung di sekitar	
10	Di dalam gedung 1 tingkat	
11	Di dalam gedung 2 tingkat	
12	Di dalam gedung 3 tingkat	

G. Desain Uji Coba Tingkat Akurasi Alamat Google Map

Tabel 3.12 Desain Uji Coba Tingkat Akurasi Alamat Google Map

Test Case ID	Tempat Uji Coba	Terdeteksi
13		
14		
15		
16		
17		

H. Desain Uji Coba Tingkat Akurasi GPS

Tabel 3.13 Desain Uji Coba Tingkat Akurasi GPS

Test Case ID	Tempat Uji Coba		Terdeteksi		Selisih
	Latitude	Longitude	Latitude	Longitude	
18					
19					
20					
21					
22					

I. Desain Uji Coba Tingkat Akurasi Voice Recognition Dalam Kebisingan

Tabel 3.14 Desain Uji Coba Tingkat Akurasi Voice Recognition

Test Case ID	Tingkat Kebisingan	Terdeteksi
18	20 db	
23	30 db	
24	40 db	
25	50 db	

26	60 db	
27	70 db	
28	80 db	
29	90 db	
30	100 db	

STIKOM SURABAYA