

Pengembangan Aplikasi Pengingat Aktivitas Berdasarkan Lokasi Berbasis Android

TUGAS AKHIR

Disusun oleh:

Dimas Rendy Febianto

3311511007

Disusun untuk memenuhi salah satu syarat kelulusan Program Diploma III



PROGRAM STUDI TEKNIK INFORMATIKA

JURUSAN TEKNIK INFORMATIKA

POLITEKNIK NEGERI BATAM

BATAM

2018

HALAMAN PENGESAHAN

Pengembangan Aplikasi Pengingat Aktivitas Berdasarkan Lokasi Berbasis Android

**Disusun oleh:
Dimas Rendy Febianto
3311511007**

Telah dikonsultasikan dengan dosen pembimbing
sebagai persyaratan untuk melaksanakan sidang Tugas Akhir II

Batam, 27 September 2018
Disetujui oleh:
Pembimbing,

Muchammad Fajri Amirul Nasrullah, S.ST., M.Sc
NIK. 117173

ABSTRAK

Pengingat aktivitas sering dilakukan secara konvensional dimana informasi terkait aktivitas yang ingin dilakukan dan lokasi aktivitas hanya tertulis secara tekstual. Namun bagi seseorang yang memiliki jadwal yang padat, cara konvensional tersebut sedikit merepotkan dan rentan terhadap kelupaan. Hal ini terjadi tidak lepas dari keterbatasan manusia dalam mengelola data yang tidak se-efektif komputer, dimana data selalu dapat diproses selama data tersebut tersimpan. Oleh karena itu, dengan semakin berkembangnya teknologi alat komunikasi seperti *smartphone*, serta didukung dengan jaringan internet yang mulai merata, maka informasi dalam bentuk *mobile reminder* akan terasa lebih efektif dan *modern* karena *smartphone* selalu dibawa oleh manusia saat ini.

Berdasarkan latar belakang di atas, maka dalam penelitian ini akan dikembangkan sebuah aplikasi *android* yang memiliki fungsi sebagai pengingat pengguna mengenai agenda aktivitas yang akan dilakukan. Yaitu, aplikasi yang memberi peringatan kepada pengguna berdasarkan waktu yang telah ditentukan, memberikan pengingat kepada pengguna jika lokasi pengguna berada dibawah 1 km dari lokasi tujuan yang diharapkan dapat membuat pengguna mengingat kembali hal apa yang ingin dilakukannya disekitar lokasi tersebut, serta memberikan direksi menuju ke lokasi tujuan yang di harapkan dapat mengurangi resiko tersesat kepada pengguna yang tidak mengetahui lokasi tujuan dengan baik. Oleh karena itu, peneliti akan memanfaatkan fitur *Google Maps API* dalam penelitian ini.

Keywords: Pengingat, *Android*, *Google Maps API*

DAFTAR ISI

HALAMAN PENGESAHAN.....	i
ABSTRAK.....	ii
DAFTAR ISI.....	iii
DAFTAR TABEL.....	vi
DAFTAR GAMBAR.....	vii
BAB I	
PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika Penulisan.....	3
BAB II	
LANDASAN TEORI.....	5
2.1 Landasan Teori.....	5
2.1.1 Android.....	5
2.1.2 Ionic.....	5
2.1.3 Location Based Service.....	6
2.1.4 <i>Google Maps API</i>	6
2.1.5 HTML5.....	7
2.1.6 <i>Haversine</i> Formula.....	7
2.1.7 Node.js.....	8
2.1.8 Javascript.....	9
2.1.9 SQLite.....	9
2.1.10 <i>Reminder</i>	10
2.2 Tinjauan Pustaka.....	1
BAB III	
ANALISIS DAN PERANCANGAN.....	12
3.1 Deskripsi Umum Sistem.....	12
3.1.1 Kebutuhan Fungsional.....	13
3.1.2 Kebutuhan Non Fungsional.....	13

3.2	<i>Usecase Diagram</i>	14
3.3	Skenario <i>Use Case</i>	14
3.3.1	Skenario Use Case Tambah Pengingat.....	14
3.3.2	Skenario Use Case Ubah Pengingat	Error! Bookmark not defined.
3.3.3	Skenario Use Case Hapus Pengingat	16
3.3.4	Skenario Use Case Lihat Lokasi Pengingat.....	16
3.4	<i>Activity Diagram</i>	17
3.4.1	<i>Activity Diagram</i> Tambah Pengingat.....	17
3.4.2	<i>Activity Diagram</i> Ubah Pengingat	18
3.4.3	<i>Activity Diagram</i> Hapus Pengingat	18
3.4.4	<i>Activity Diagram</i> Lihat Lokasi Pengingat.....	19
3.5	<i>Sequence Diagram</i>	20
3.5.1	<i>Sequence Diagram</i> Tambah Pengingat.....	20
3.5.2	<i>Sequence Diagram</i> Ubah Pengingat	21
3.5.3	<i>Sequence Diagram</i> Hapus Data Pengingat	21
3.5.4	<i>Sequence Diagram</i> Lihat Lokasi Pengingat.....	22
3.6	Perancangan Antarmuka	23
3.6.1	Perancangan Antarmuka Halaman Utama	23
3.6.2	Perancangan Antarmuka Tambah Pengingat.....	24
3.6.3	Perancangan Antarmuka Ubah Pengingat	25
3.6.4	Perancangan Antarmuka Hapus Pengingat	26
3.6.5	Perancangan Antarmuka Lihat Lokasi Pengingat	27
BAB IV		
IMPLEMENTASI DAN PENGUJIAN SISTEM		
4.1	Implementasi <i>Database</i>	28
4.2	Implementasi Antarmuka.....	28
4.2.1	Implementasi Antarmuka Halaman Utama	28
4.2.2	Implementasi Antarmuka Halaman Tambah Pengingat.....	30
4.2.3	Implementasi Antarmuka Halaman Ubah Pengingat	32
4.2.4	Implementasi Antarmuka Halaman Hapus Pengingat.....	34
4.2.5	Implementasi Antarmuka Halaman Lihat Lokasi Pengingat.....	35

4.2.6 Implementasi Notifikasi	36
4.3 Pengujian	36
BAB V	
KESIMPULAN DAN SARAN	38
5.1 Kesimpulan	38
5.2 Saran	38
Daftar Pustaka	39

DAFTAR TABEL

Tabel 2.1 Tinjauan Pustaka	11
Tabel 3.1 <i>Use Case</i> Tambah Pengingat	14
Tabel 3.2 <i>Use Case</i> Ubah Pengingat	14
Tabel 3.3 <i>Use Case</i> Hapus Pengingat	15
Tabel 3.4 <i>Use Case</i> Lihat Lokasi Pengingat	16
Tabel 3.5 Perancangan Antarmuka Halaman Utama	23
Tabel 3.6 Perancangan Antarmuka Tambah Pengingat	24
Tabel 3.7 Perancangan Antarmuka Ubah Pengingat	26
Tabel 3.8 Perancangan Antarmuka Hapus Pengingat	26
Tabel 3.9 Perancangan Antarmuka Lihat Lokasi Pengingat	27
Tabel 4.1 Tabel Pengingat	28

DAFTAR GAMBAR

Gambar 3.1 Deskripsi Umum Sistem.....	12
Gambar 3.2 <i>Usecase</i> diagram.....	14
Gambar 3.3 <i>Activity Diagram</i> Tambah Pengingat	17
Gambar 3.4 <i>Activity Diagram</i> Ubah Pengingat.....	18
Gambar 3.5 <i>Activity Diagram</i> Hapus Pengingat	19
Gambar 3.6 <i>Activity Diagram</i> Lihat Lokasi Pengingat	20
Gambar 3.7 <i>Sequence</i> Tambah Pengingat.....	20
Gambar 3.8 <i>Sequence</i> Ubah Pengingat	21
Gambar 3.9 <i>Sequence</i> Hapus Pengingat.....	22
Gambar 3.10 <i>Sequence</i> Lihat Lokasi Pengingat.....	22
Gambar 3.11 Perancangan Antarmuka Halaman Utama.....	23
Gambar 3.12 Perancangan Antarmuka Tambah Pengingat.....	24
Gambar 3.13 Perancangan Antarmuka Ubah Pengingat	25
Gambar 3.14 Perancangan Antarmuka Hapus Pengingat	26
Gambar 3.15 Perancangan Antarmuka Lihat Lokasi Pengingat.....	27
Gambar 4.1 Implementasi Antarmuka Halaman Utama	29
Gambar 4.2 <i>Source Code</i> Halaman Utama	29
Gambar 4.3 Implementasi Antarmuka Halaman Tambah Pengingat	30
Gambar 4.4 <i>Source Code</i> Halaman Tambah Pengingat.....	31
Gambar 4.5 Implementasi Antarmuka Halaman Ubah Pengingat	32
Gambar 4.6 <i>Source Code</i> Halaman Ubah Pengingat	33
Gambar 4.7 Implementasi Antarmuka Halaman Hapus Pengingat.....	34
Gambar 4.8 <i>Source Code</i> Halaman Hapus Pengingat.....	34
Gambar 4.9 Implementasi Antarmuka Lihat Lokasi Pengingat	35
Gambar 4.10 <i>Source Code</i> Lihat Lokasi Pengingat	35
Gambar 4.11 Implementasi Notifikasi	36

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kebutuhan akan informasi merupakan salah satu aspek penting dalam kehidupan manusia di era globalisasi sekarang ini. Informasi adalah hasil dari pengolahan data yang memberikan arti & manfaat tertentu bagi orang yang menerimanya (Azhar Susanto, 2017:3). Salah satu wujud pentingnya informasi adalah sebagai pengingat aktivitas yang ingin dilakukan. Biasanya, pengingat aktivitas dilakukan dengan metode konvensional dimana informasi terkait aktivitas yang ingin dilakukan dan lokasi aktivitas tersebut hanya ditulis secara tekstual. Akan tetapi, penggunaan metode ini tidak dapat diterapkan oleh semua orang.

Lupa merupakan sebuah istilah yang sudah tidak asing ditelinga masyarakat. Setiap orang dalam kehidupannya sehari-hari mengalami fenomena ini. Seseorang dapat lupa akan suatu peristiwa atau kejadian di masa lampau, maupun lupa terhadap sesuatu yang akan dilakukan. Fenomena ini dapat terjadi pada siapapun juga, tak peduli apakah orang itu anak-anak, remaja, orang tua, guru, pejabat, profesor, petani, dan sebagainya (Syaiful Bahri Djamarah, 2008: 206). Salah satu contoh kerugian dari terjadinya lupa adalah kerugian waktu seperti seseorang ingin membeli makanan kucing di sebuah *mall*, ia berencana membelinya saat akhir pekan, namun saat akhir pekan tiba ia juga memiliki rencana untuk menonton di *mall* tersebut, maka besar kemungkinan orang tersebut tidak bisa mengingat rencana awalnya untuk membeli makanan kucing karena terbawa kegiatan yang akan dilakukannya di *mall* tersebut yaitu menonton film, sehingga ia lupa membeli makanan kucing yang awalnya ia rencanakan dan harus membelinya di akhir pekan berikutnya.

Fenomena lupa sebenarnya sudah coba diantisipasi oleh masyarakat, yaitu dengan membawa kertas catatan saat berpergian. Bagaimanapun, kelalaian manusia seperti seringnya kertas tersebut terselip atau hilang membuat metode ini

memiliki tingkat keefektifan yang rendah. Cara lain yang juga sering digunakan oleh masyarakat adalah dengan membuat *notes* pada *smartphone*. Sayangnya, *notes* tersebut pun sering terlupa karena terbawa kegiatan yang dilakukan dan jika tidak ada pemicu pengingatnya maka hal ini pun sering terlewatkan. Hal ini terjadi tidak lepas dari keterbatasan otak manusia dalam mengelola data yang tidak seefektif komputer, dimana data selalu dapat diproses selama data tersebut tersimpan. Oleh karena itu, dengan semakin berkembangnya teknologi alat komunikasi seperti *smartphone*, serta didukung dengan jaringan internet yang mulai merata, maka informasi dalam bentuk *mobile reminder* akan terasa lebih efektif karena *smartphone* selalu dibawa oleh manusia saat ini.

Berdasarkan latar belakang di atas, maka dalam penelitian ini akan dikembangkan sebuah aplikasi *android* yang memiliki fungsi sebagai pengingat pengguna mengenai agenda aktivitas yang akan dilakukan. Yaitu, aplikasi yang memberi peringatan kepada pengguna berdasarkan waktu yang telah ditentukan, memberikan pengingat kepada pengguna berdasarkan lokasi jika lokasi pengguna berada disekitar 1 km dari lokasi tujuan yang diharapkan dapat membuat pengguna mengingat kembali hal apa yang ingin dilakukannya disekitar lokasi tersebut, serta memberikan direksi menuju ke lokasi tujuan yang di harapkan dapat mengurangi resiko tersesat kepada pengguna yang tidak mengetahui lokasi tujuan dengan baik. Oleh karena itu, peneliti akan memanfaatkan fitur *Google Maps API* dalam penelitian ini.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, berikut adalah rumusan masalah dalam penelitian ini:

1. Bagaimana membangun sebuah aplikasi yang dapat meminimalisir resiko terjadinya kelupaan?

1.3 Tujuan

Tujuan yang ingin di capai melalui penelitian ini adalah sebagai berikut:

1. Membangun aplikasi yang dapat mengurangi resiko terjadinya kelupaan akan suatu kegiatan ataupun barang yang diinginkan.

1.4 Manfaat

Adapun manfaat tugas akhir ini adalah sebagai berikut:

1. Meminimalisir permasalahan akan kelupaan yang sering terjadi di masyarakat.
2. Membantu pengguna dengan memberikan informasi dan mengingatkan tentang aktivitas yang ingin dilakukan.

1.5 Batasan Masalah

Untuk membatasi cakupan pembahasan masalah pada Tugas Akhir ini, maka diberikan batasan-batasan sebagai berikut:

1. Aplikasi hanya memberikan pemberitahuan sebatas apa yang sudah di masukkan oleh pengguna.
2. Aplikasi harus dibuka terlebih dahulu agar fungsi pengingat berdasarkan lokasi dapat berfungsi.

1.6 Sistematika Penulisan

Sistematika dalam penulisan Tugas Akhir ini, meliputi :

BAB I : **PENDAHULUAN**, bagian ini dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat dan sistematika penulisan.

BAB II : **LANDASAN TEORI**, bagian ini diuraikan dan dijelaskan mengenai dasar teori.

BAB III : **ANALISIS dan PERANCANGAN**, bagian ini dijelaskan mengenai langkah-langkah penyelesaian masalah.

BAB IV : **IMPLEMENTASI dan PENGUJIAN SISTEM**, bagian ini memuat uraian langkah implementasi dan pengujian/validasi serta pembahasan tentang hasil uji coba dari aplikasi.

BAB V : **KESIMPULAN dan SARAN**, bagian ini memuat simpulan yang merupakan rangkuman dari hasil analisis kinerja pada bagian sebelumnya serta berisi saran-saran pengembangan dari penelitian yang dibuat dan aspek yang belum terselesaikan.

BAB II

LANDASAN TEORI

2.1 Landasan Teori

2.1.1 Android

Menurut Nazaruddin Safaat H (2012:1) *Android* adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka.

Sedangkan menurut Yosef Murya (2014:3) *Android* adalah sistem operasi berbasis Linux yang digunakan untuk telepon seluler seperti telepon pintar (*smartphone*) dan komputer tablet (PDA).

2.1.2 Ionic

Ionic adalah sebuah *framework* aplikasi *mobile* berbasis HTML5 yang dapat digunakan untuk mengembangkan aplikasi *mobile* dengan teknologi *web* seperti HTML, CSS, dan Javascript. Dengan menggunakan Ionic, para *developer web* bisa membuat aplikasi lintas *platform* seperti untuk *Android* dan *iOS*. Dengan menggunakan Ionic, para *developer web* tidak perlu belajar bahasa pemrograman Java, Objective C, atau C# untuk membuat aplikasi *mobile* karena mereka cukup menggunakan ilmu mereka di bahasa pemrograman *web*.

Ionic mempunyai kelebihan diantaranya bersifat *open source* dan juga menggunakan teknologi *web* terbaru. Ionic juga menggunakan Angular JS untuk implementasi logikanya. Dengan menggunakan Angular JS aplikasi yang dibuat menggunakan Ionic akan dapat berjalan sangat cepat seperti halnya aplikasi *native*.

Ionic hanyalah sebuah *framework*, maka untuk mengubah kode yang telah dibuat menjadi aplikasi Android atau iOS perlu menggunakan tool tambahan seperti *Cordova / Phonegap*.

2.1.3 Location Based Service

Location Based Service (LBS) adalah “sekumpulan aplikasi yang mengeksplorasi pengetahuan /informasi dari lokasi geografis perangkat *mobile* untuk mendapatkan layanan berdasarkan informasi tersebut”.

Pemanfaatan LBS memudahkan pengguna perangkat *mobile* mengatur dan memilih layanan sesuai kebutuhan dan dapat dimanfaatkan untuk memberikan berbagai layanan seperti informasi kondisi lingkungan (kemacetan lalu lintas, cuaca, lokasi fasilitas umum terdekat), maupun promosi produk dan jasa.

Pemanfaatan LBS pada sistem operasi perangkat *mobile* dimungkinkan dengan adanya dukungan dua unsur utama berikut:

1. *Location Manager (API Maps)*

Menyediakan *tool* atau *source* untuk *Location Based Services Application Program Interface (API Maps)* yang menyediakan fasilitas untuk menampilkan, mengubah maps atau peta beserta fitur lainnya seperti tampilan satelit, lokasi, dan jalan suatu tempat. Paket ini berada pada *com.google.android.map*.

2. *Location Provider (API Location)*

Menyediakan teknologi pencarian lokasi yang digunakan oleh perangkat. *API Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi secara *real-time*. *API Location* berada pada paket *android.location*. Dengan *Location Manager*, kita dapat menentukan lokasi kita saat ini berdasarkan gerakan, serta kedekatan dengan lokasi tertentu dengan mendeteksi perpindahan.

2.1.4 Google Maps API

Google Maps API adalah sebuah layanan yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Maps* dalam

mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi *enterprise* di dalam sebuah *website* maupun di dalam aplikasi *mobile*.

Pengguna dapat memanfaatkan layanan-layanan yang ditawarkan oleh *Google Maps* setelah melakukan registrasi dan mendapatkan *Google Maps API Key*. *Google* menyediakan layanan ini secara gratis kepada pengguna di seluruh dunia.

2.1.5 HTML5

Menurut Iqbal, Husni dan Studiawan (2012), HTML5 (*Hypertext Markup Language version 5*) adalah sebuah bahasa yang menstrukturkan isi dari *World Wide Web*, sebuah teknologi utama pada internet. Standar HTML5 menyempurnakan elemen-elemen lama yang terdapat pada standar sebelumnya, menambahkan elemen-elemen yang lebih semantik dan menambahkan fitur-fitur baru untuk mendukung pembuatan aplikasi *web* yang lebih kompleks.

Menurut R.H. Sianipar (2015) HTML5 (*Hypertext Markup Language version 5*) adalah sebuah *markup* untuk menstrukturkan dan menampilkan isi dari halaman *web*. HTML5 merupakan salah satu karya *World Wide Web Consortium*, W3C untuk mendefinisikan sebuah Bahasa *markup* tunggal yang dapat ditulis dengan cara HTML ataupun XHTML. HTML5 merupakan jawaban atas pengembangan HTML 4.01 dan XHTML 1.1 yang selama ini berjalan terpisah, dan diimplementasikan secara berbeda-beda oleh banyak perangkat lunak *web*.

2.1.6 Haversine Formula

Formula *Haversine* adalah persamaan penting dalam sistem navigasi, yang dalam penggunaannya formula *haversine* akan menghasilkan jarak terpendek antara dua titik. Formula ini pertama kali ditemukan oleh Jamez Andrew di tahun 1805, dan digunakan pertama kali oleh Josef de Mendoza

y Ríos di tahun 1801. Istilah *haversine* ini sendiri diciptakan pada tahun 1835 oleh Prof. James Inman. Josef de Mendoza y Ríos menggunakan *haversine* pertama kali dalam penelitiannya tentang “Masalah Utama Astronomi *Nautical*”.

Formula *Haversine* adalah persamaan yang digunakan dalam navigasi, yang memberikan jarak lingkaran besar antara dua titik pada permukaan bumi berdasarkan *latitude* dan *longitude*. Formula *Haversine* dalam penelitian ini berfungsi untuk menghitung jarak lokasi tujuan dengan lokasi pengguna berada.

Penggunaan rumus ini mengasumsikan pengabaian efek *ellipsoidal*, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Berikut adalah rumus *haversine*:

$$\begin{aligned}\Delta\text{lat} &= \text{lat2} - \text{lat1} \\ \Delta\text{long} &= \text{long2} - \text{long1} \\ a &= \sin^2(\Delta\text{lat}/2) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \sin^2(\Delta\text{long}/2) \\ c &= 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c\end{aligned}$$

Dimana :

- R = jari-jari bumi sebesar 6371(km)
- Δlat = besaran perubahan latitude
- Δlong = besaran perubahan longitude
- C = kalkulasi perpotongan sumbu
- d = jarak (km)
- 1 derajat = 0.0174532925 radian

2.1.7 Node.js

Menurut Iqbal, Husni dan Studiawan (2012) Node.js adalah sistem perangkat lunak yang didesain untuk pengembangan aplikasi *web*. Tidak seperti kebanyakan bahasa JavaScript yang dijalankan pada *web browser*, Node.js dieksekusi sebagai aplikasi *server*. Aplikasi ini terdiri dari V8 JavaScript *Engine* buatan *Google* dan beberapa modul bawaan yang terintegrasi. Dalam buku Teixeira (2013), *server side javascript* sudah beredar dari beberapa tahun lalu. Dengan menggunakan Node.js kita dapat membuat

model *programming* yang mudah dengan skala *server* yang cukup besar dan mudah.

2.1.8 Javascript

Menurut Wahana Komputer (2009:1) JavaScript merupakan bahasa pemrograman untuk memberikan kemampuan tambahan kedalam bahasa pemrograman HTML dengan mengizinkan pengeksekusian perintah-perintah pada sisi *client*, bukan pada sisi *server* dokumen *web*. Pada hakikatnya, bahasa pemrograman JavaScript berisi *script* yang pemasangannya terselip disebuah dokumen HTML, sehingga bahasa JavaScript ini tidak memerlukan sebuah kompilator atau penerjemah khusus untuk mengeksekusinya.

Menurut Lani Sidharta (1997: 1) JavaScript adalah *script* sederhana untuk menyusun aplikasi-aplikasi internet untuk *client* dan *server*.

Menurut R.H. Sianipar (2015: 1) JavaScript merupakan bahasa *script* populer yang dipakai untuk menciptakan halaman *web* yang dapat berinteraksi dengan pengguna dan dapat merespon *event* yang terjadi pada halaman. JavaScript merupakan penghubung yang menyatukan halaman-halaman *web*.

2.1.9 SQLite

SQLite adalah salah satu *software* basis data yang tertanam yang sangat populer, kombinasi SQL *interface* dan penggunaan *memory* yang sangat sedikit dengan kecepatan yang sangat cepat. SQLite di *android* termasuk dalam *Android runtime*, sehingga setiap versi dari *android* dapat membuat *database* dengan SQLite. SQLite sendiri terkenal sebagai *database* yang *open-source*, *standalone* SQL *database*, berukuran kecil, tidak membutuhkan administrasi, tanpa server, tanpa file konfigurasi dan juga telah digunakan pada banyak aplikasi terkenal. Pada *Android*, *database* yang dibuat untuk suatu aplikasi hanya bisa diakses oleh aplikasi itu sendiri, aplikasi lain tidak akan bisa mengaksesnya kecuali jika ingin berbagi *datasenya* yaitu dengan menggunakan *content provider*. *Database* SQLite merupakan sebuah file yang dapat diambil, dipindahkan ke sistem lain. (Ayu Kusuma, 2013).

2.1.10 *Reminder*

Reminder atau pengingat bisa dikatakan sebagai sebuah sistem yang berfungsi untuk memberi tahu pada hari/waktu itu ada sebuah kegiatan atau hal yang harus dilakukan. *Reminder* biasanya berkaitan erat dengan alarm dan agenda. Alarm pada umumnya untuk memberi peringatan kepada pengguna bahwa ada suatu kegiatan pada waktu yang telah ditentukan sebelum alarm itu berbunyi. Biasanya sebelum mengatur pengingat menggunakan *reminder*, dirancang dulu sebuah jadwal. Pengertian jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau sistem kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan kedalam jadwal. Cara kerja sistem *reminder* adalah selalu mencocokkan jam yang sudah tersimpan di *database* sistem. Jika jam sudah menunjukkan seperti yang ada pada *database*, maka aplikasi akan memberikan notifikasi seperti yang diharapkan agar memberikan pengingat kepada pengguna.

2.2 Tinjauan Pustaka

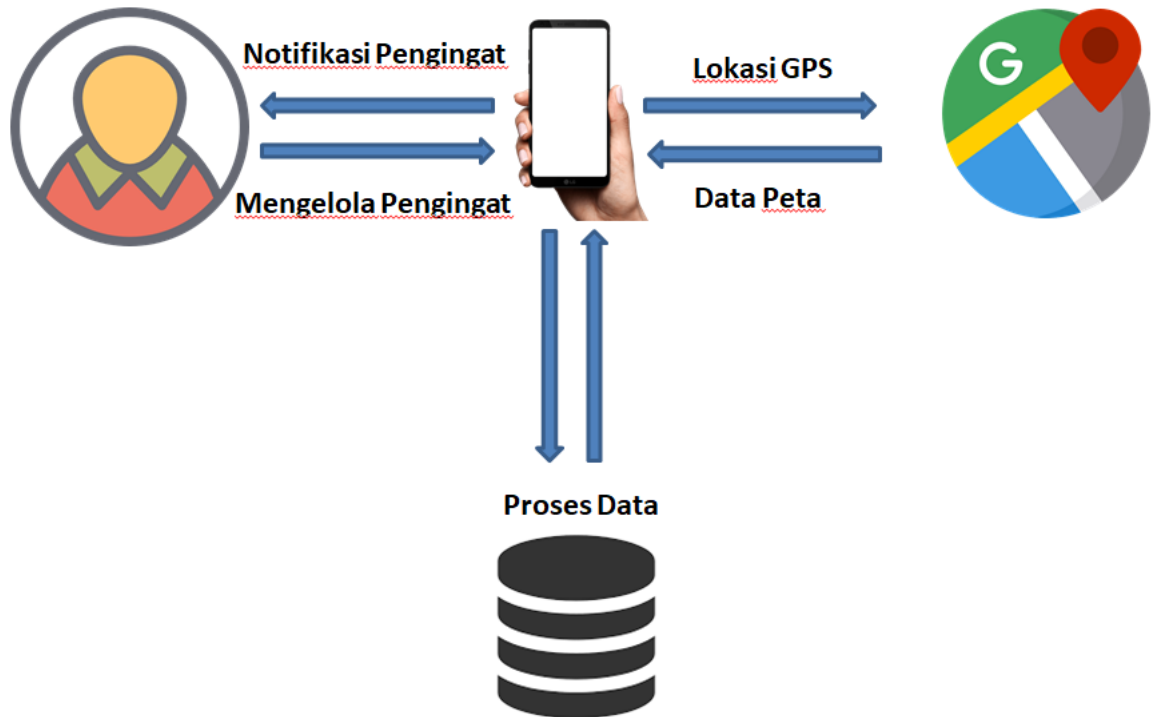
Tabel 2.1 Tinjauan Pustaka

No	Judul	Peneliti dan Tahun	Teknologi yang digunakan	Keterangan	Perbedaan
1		Dimas 2017	GSM, WIFI, GPS, API Google Maps	Memberikan reminder kepada pengguna jika pengguna sudah berada di sekitar lokasi yang sudah didaftarkan sebelumnya oleh pengguna sendiri	-
2	<i>Location Based Reminder Using GPS for Mobile (Android)</i>	Priyanka Shah, Ruta Gadgil, Neha Tamhankar 2012	GSM, GPS, Google Maps, KDDI-P	Menggunakan skema “ <i>client-server</i> ” berbasis <i>web</i> dengan menggunakan apache, MySQL, dan PHP.	Menggunakan J2ME untuk <i>Java Runtime</i> dan KDDI-P untuk GPS.
3	<i>Reminder Based On The User’s Location</i>	Dragan Perakovic, Vladimir Remenar	GSM, GPS,	Pengguna harus tepat berada di koordinat yang sudah ditentukan sebelumnya agar <i>reminder</i> dapat berfungsi	Menggunakan koordinat sebagai acuan <i>reminder</i> .
4	<i>Location Based Reminder Using Android and Google Maps</i>	Neelu. L, Aruna Kumara.B, Shashidhar. V, Bharath J 2015	GPS, API Google Maps, Proximity, GSM,	Menggunakan <i>proximity</i> untuk memberikan pengingat jika sudah berada di sekitar lokasi	Menggunakan <i>proximity</i> .

BAB III

ANALISIS DAN PERANCANGAN

3.1 Deskripsi Umum Sistem



Gambar 3.1 Deskripsi Umum Sistem

Deskripsi dari proses kerja sistem pada gambar 3.1 adalah sebagai berikut :

1. Pengguna dapat melakukan pengelolaan data pengingat yaitu membuat, mengubah, dan menghapus data pengingat. Data pengingat yang telah dibuat akan disimpan pada SQLite *database*. Data yang disimpan adalah judul pengingat, deskripsi pengingat, tanggal, jam, dan lokasi pengingat. Lokasi pengingat akan disimpan dan disinkronisasikan dengan *Google Maps API*. Aplikasi akan memberikan pengingat berdasarkan waktu yang dibuat oleh pengguna, serta berdasarkan lokasi saat pengguna berada di bawah 1 km dari lokasi tujuan.
2. GPS dan *Internet* harus dalam keadaan aktif jika pengguna ingin memilih lokasi dan dapat menggunakan fitur *map* dan pengingat lokasi sehingga

aplikasi secara langsung memeriksa lokasi koordinat pengguna dengan lokasi tujuan.

3.1.1 Kebutuhan Fungsional

F-001 User dapat melakukan tambah pengingat aktivitas

F-002 User dapat melakukan ubah pengingat aktivitas

F-003 User dapat melakukan hapus pengingat aktivitas

F-004 Sistem memberikan notifikasi berdasarkan pengingat aktivitas yang telah dibuat

3.1.2 Kebutuhan Non Fungsional

NF-001 Bahasa yang digunakan dalam sistem ini adalah bahasa Indonesia

3.1.2.1 Analisa Kebutuhan Perangkat Keras

Spesifikasi perangkat keras yang dibutuhkan dalam Pengembangan Aplikasi Pengingat Aktivitas Berdasarkan Lokasi Berbasis Android adalah

1. Laptop AMD A12 Gen 7th.
2. 8GB RAM.
3. 1TB HDD.
4. VGA AMD Radeon R7 Graphics.

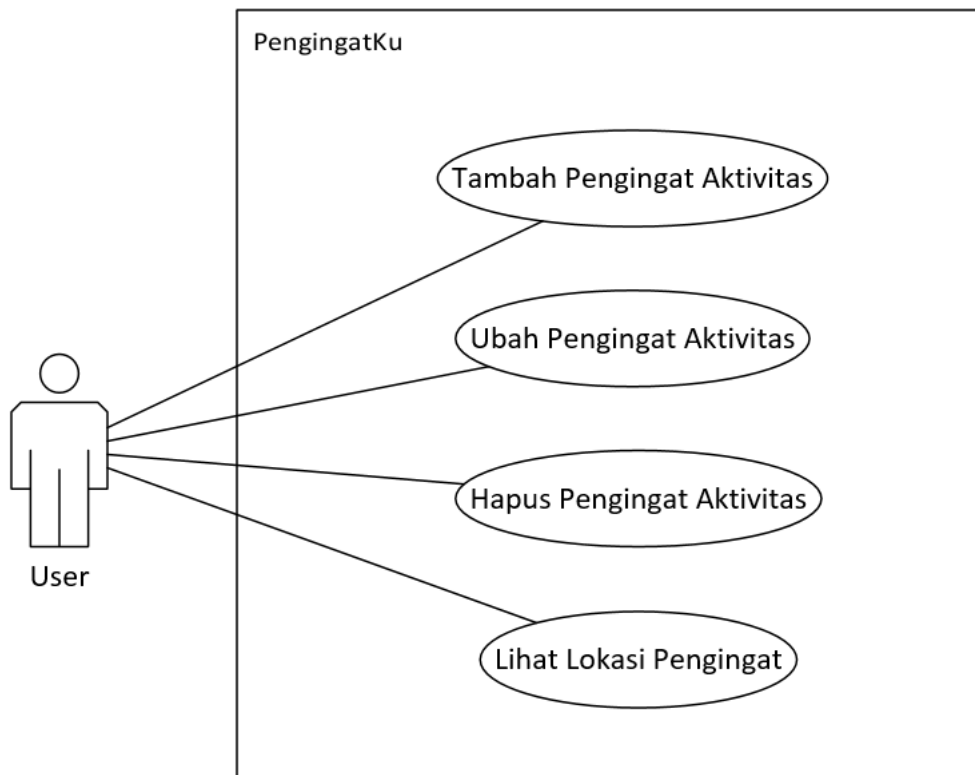
3.1.2.2 Analisa Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam Pengembangan Aplikasi Pengingat Aktivitas Berdasarkan Lokasi Berbasis Android adalah :

1. Visual Studio Code
2. Ionic Framework
3. Google Chrome.

3.2 Usecase Diagram

Usecase diagram merupakan skenario interaksi antara pengguna dengan sistem. Usecase menggambarkan hubungan antara *actor* dan kegiatan yang dilakukan di dalam sistem. Usecase utama dari sistem ini adalah:



Gambar 3.2 Usecase Diagram

Gambar 3.2 menjelaskan bahwa pada aplikasi ini hanya terdapat 1 pengguna sehingga tidak memerlukan proses daftar dan *login* untuk dapat menggunakan aplikasi ini. Terdapat 4 kegiatan yaitu menambah pengingat, mengubah pengingat, menghapus pengingat, dan melihat lokasi pengingat yang akan ditampilkan pada *maps*.

3.3 Skenario Use Case

3.3.1 Skenario Use Case Tambah Pengingat

Tabel 3.1 Use Case Tambah Pengingat

Nama Use Case	Tambah Data Pengingat
---------------	-----------------------

Deskripsi	Pengguna dapat menambah data pengingat
Aktor	Pengguna
Kondisi Awal	Pengguna sudah di dalam menu Utama
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih tombol tambah untuk menambah data pengingat 2. Pengguna mengisi data pengingat 3. Sistem menyimpan data yang telah dimasukkan.
Skenario alternatif	-
Kondisi Akhir	Pengguna sudah menambah data pengingat

Berdasarkan *use case* tambah pengingat aktivitas pada Gambar 3.2, terdapat skenario *use case* untuk melakukan proses tambah pengingat yang dijelaskan lebih detail pada Tabel 3.1.

3.3.2 Skenario *Use Case* Ubah Pengingat

Tabel 3.2 *Use Case* Ubah Pengingat

Nama <i>Use Case</i>	Ubah Data Pengingat
Deskripsi	Pengguna dapat mengubah data pengingat
Aktor	Pengguna
Kondisi Awal	Pengguna sudah di dalam menu Utama
Skenario	<ol style="list-style-type: none"> 1. Pengguna memilih tombol ubah saat menggesar <i>listview</i> dari data pengingat yang ingin diubah. 2. Pengguna mengubah data pengingat 3. Sistem menyimpan data yang telah diubah.
Skenario alternatif	-
Kondisi Akhir	Pengguna sudah mengubah data pengingat

Berdasarkan *use case* ubah pengingat aktivitas pada Gambar 3.2, terdapat skenario *use case* untuk melakukan proses ubah pengingat yang dijelaskan lebih detail pada Tabel 3.2.

3.3.3 Skenario Use Case Hapus Pengingat

Tabel 3.3 Use Case Hapus Pengingat

Nama Use Case	Hapus Data Pengingat
Deskripsi	Pengguna dapat menghapus data pengingat
Aktor	Pengguna
Kondisi Awal	Pengguna sudah di dalam menu Utama
Skenario	1. Pengguna memilih tombol hapus saat menggesar <i>listview</i> dari data pengingat yang ingin dihapus. 2. Data terhapus
Skenario alternatif	-
Kondisi Akhir	Data pengingat pengguna sudah terhapus.

Berdasarkan *use case* hapus pengingat aktivitas pada Gambar 3.2, terdapat skenario *use case* untuk melakukan proses hapus pengingat yang dijelaskan lebih detail pada Tabel 3.3.

3.3.4 Skenario Use Case Lihat Lokasi Pengingat

Tabel 3.4 Use Case Lihat Lokasi Pengingat

Nama Use Case	Lihat Lokasi Pengingat
Deskripsi	Pengguna melihat lokasi yang telah dipilih, dan mendapatkan direksi menuju ke lokasi tujuan
Aktor	Pengguna
Kondisi Awal	Pengguna sudah di dalam menu Utama
Skenario	1. Pengguna memilih tombol pin pada data dari lokasi yang ingin dilihat pada halaman utama. 2. Pengguna memilih lokasi yang ditampilkan pada <i>Google maps</i> 3. Pengguna memilih tombol arah pada <i>Google maps</i>
Skenario alternatif	-

Kondisi Akhir	Lokasi Pengingat ditampilkan
---------------	------------------------------

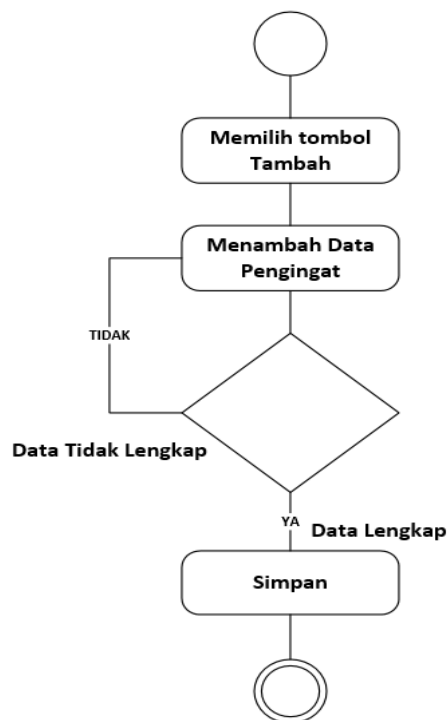
Berdasarkan *use case* lihat lokasi pengingat aktivitas pada Gambar 3.2, terdapat skenario *use case* untuk melakukan proses lihat lokasi pengingat yang dijelaskan lebih detail pada Tabel 3.4.

3.4 Activity Diagram

Activity Diagram adalah diagram yang menggambarkan aliran kerja dari setiap proses yang ada pada aplikasi.

3.4.1 Activity Diagram Tambah Pengingat

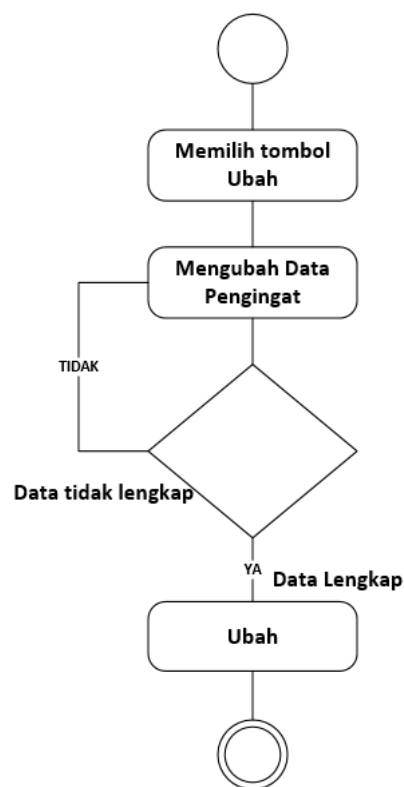
Alur proses kerja pada *activity diagram* tambah pengingat adalah pengguna memilih tombol tambah pengingat, lalu proses selanjutnya yaitu mengisi data pengingat. Semua data harus diisi oleh pengguna, jika tidak maka pengguna akan diminta untuk mengisi data ulang seperti yang disajikan pada Gambar 3.3.



Gambar 3.3 *Activity Diagram* Tambah Pengingat

3.4.2 Activity Diagram Ubah Pengingat

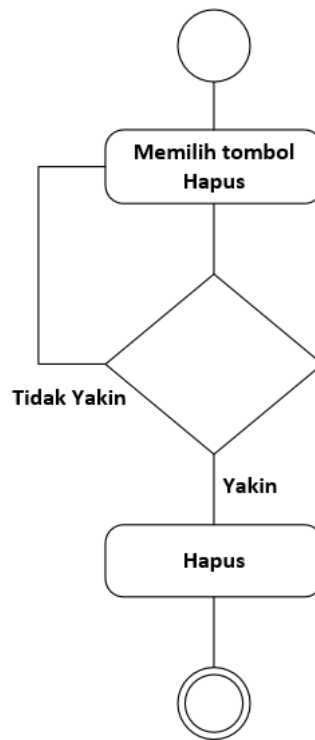
Alur proses kerja pada *activity* diagram ubah data pengingat yang sebelumnya telah ditambahkan oleh pengguna adalah pengguna menggeser *listview* sehingga akan tampil tombol ubah, lalu proses selanjutnya yaitu mengubah data pengingat. Semua data harus diisi oleh pengguna, jika tidak maka pengguna akan diminta untuk mengisi data ulang seperti yang disajikan pada Gambar 3.4



Gambar 3.4 Activity Diagram Ubah Pengingat

3.4.3 Activity Diagram Hapus Pengingat

Alur proses kerja pada *activity* diagram menghapus data pengingat yang sebelumnya telah dibuat oleh pengguna adalah pengguna menggeser *listview* sehingga akan tampil tombol hapus. Saat pengguna memilih tombol hapus, maka sistem akan memberikan peringatan kepada pengguna apakah pengguna yakin mau menghapus data atau tidak seperti yang ditampilkan pada Gambar 3.5.



Gambar 3.5 *Activity Diagram* Hapus Pengingat

3.4.4 **Activity Diagram Lihat Lokasi Pengingat**

Alur proses kerja pada *activity* diagram lihat lokasi pengingat adalah menekan tombol pin pada data yang ingin dilihat lokasinya. Lalu sistem akan menampilkan *map* beserta lokasi tujuan pengingat. Untuk mendapatkan direksi menuju tempat tujuan, pengguna harus menekan tombol arah yang berada dibawah kanan pada aplikasi seperti yang disajikan pada Gambar 3.6.

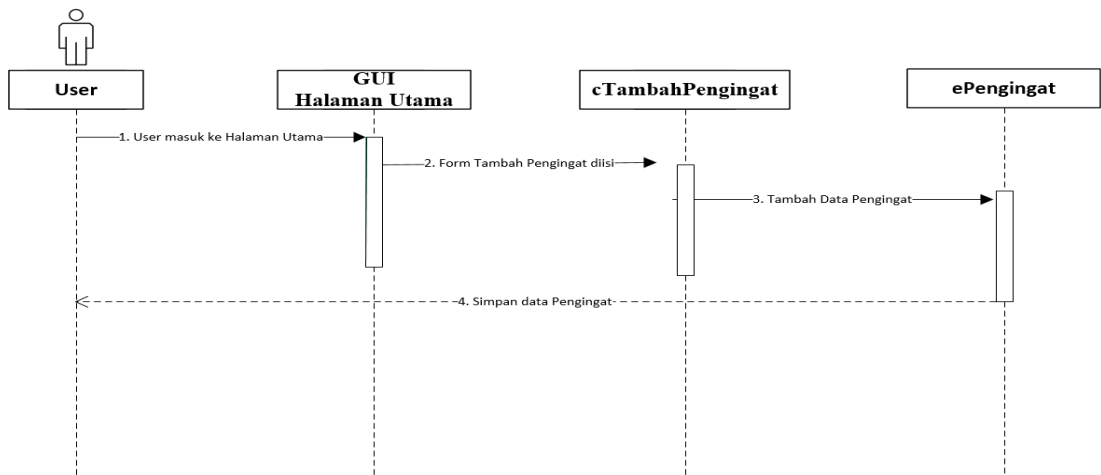


Gambar 3.6 *Activity Diagram* Lihat Lokasi Peningat

3.5 Sequence Diagram

3.5.1 Sequence Diagram Tambah Peningat

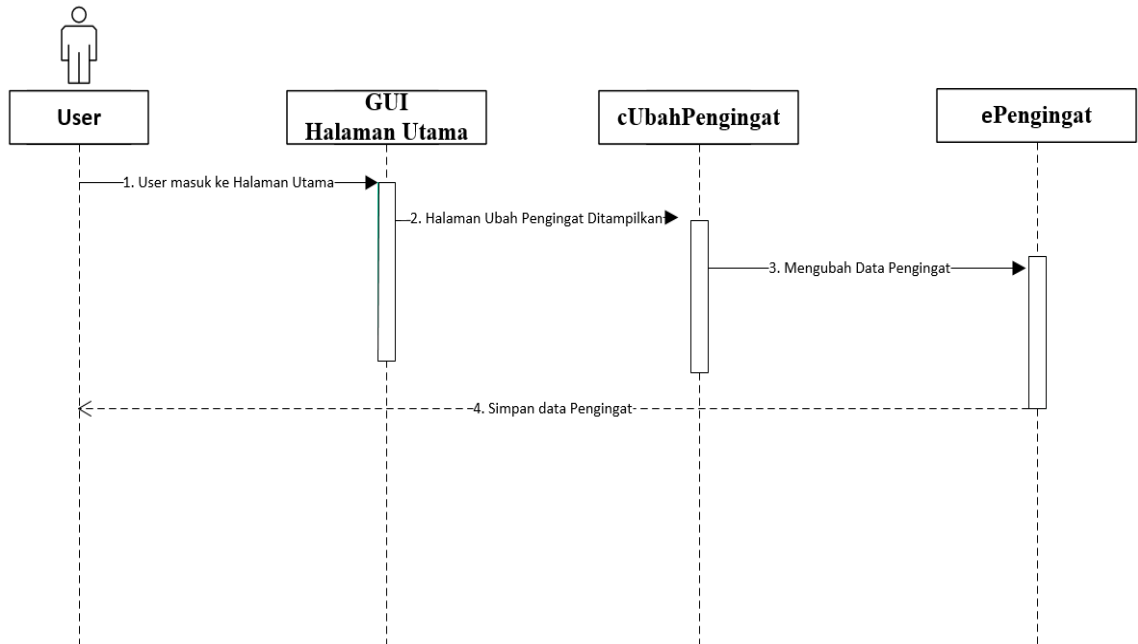
Pengguna telah berhasil masuk ke dalam aplikasi dan ke halaman utama. Pengguna dapat menambahkan data peningat lalu disimpan seperti yang disajikan pada Gambar 3.7.



Gambar 3.7 *Sequence* Tambah Peningat

3.5.2 Sequence Diagram Ubah Pengingat

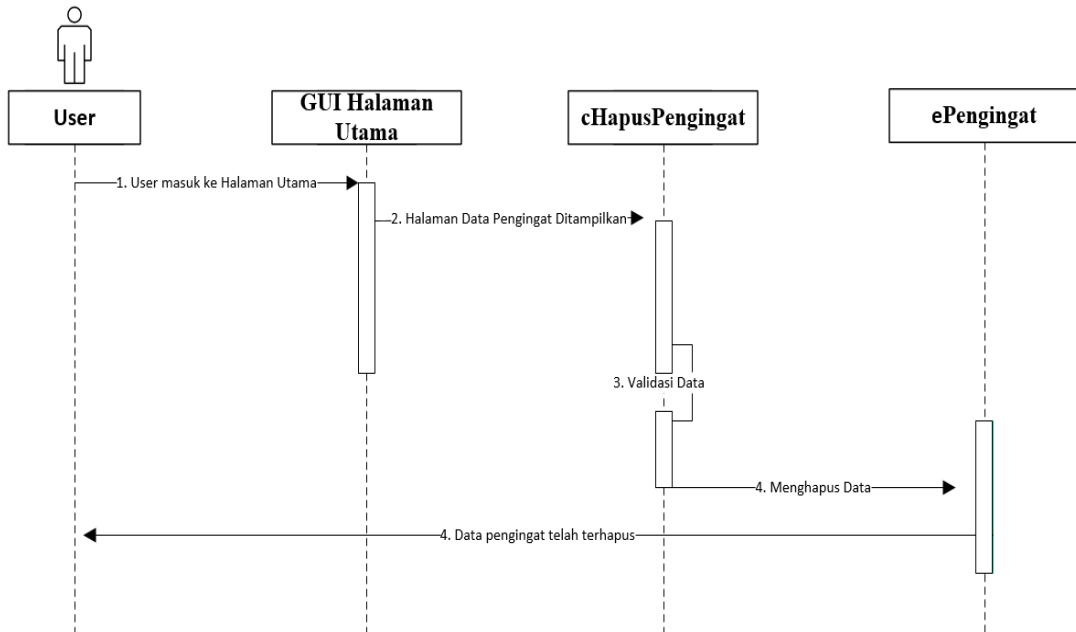
Pengguna telah berhasil masuk ke dalam aplikasi dan ke halaman utama. Pengguna dapat mengubah data pengingat yang telah dibuat lalu disimpan seperti yang disajikan pada Gambar 3.8



Gambar 3.8 Sequence Ubah Pengingat

3.5.3 Sequence Diagram Hapus Data Pengingat

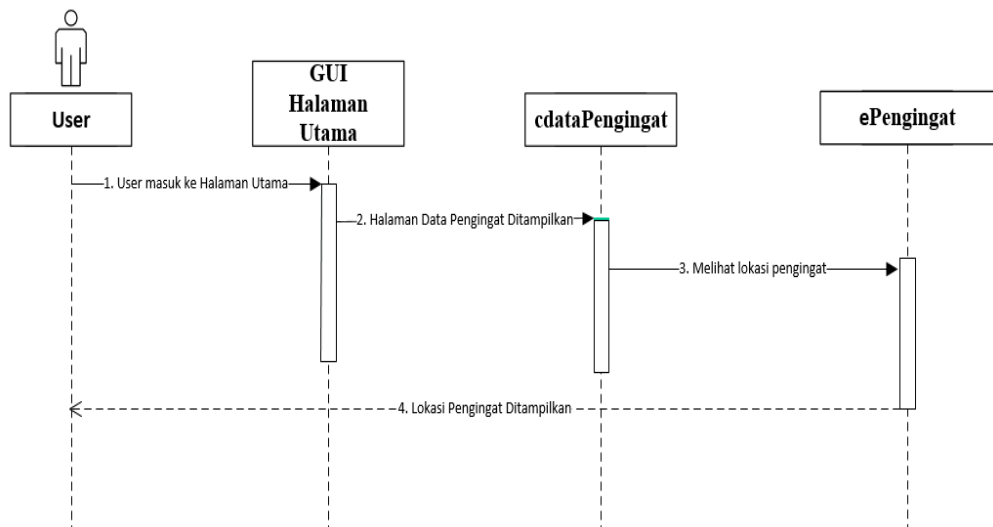
Pengguna telah berhasil masuk ke dalam aplikasi dan ke halaman utama. Pengguna dapat menghapus data pengingat yang telah dibuat lalu disimpan seperti yang disajikan pada Gambar 3.9.



Gambar 3.9 *Sequence* Hapus Peningat

3.5.4 *Sequence* Diagram Lihat Lokasi Peningat

Pengguna telah berhasil masuk ke dalam aplikasi dan ke halaman utama. Pengguna dapat melihat lokasi pengingat pada list pengingat di halaman utama seperti yang disajikan pada Gambar 3.10.



Gambar 3.10 *Sequence* Lihat Lokasi Peningat

3.6 Perancangan Antarmuka

3.6.1 Perancangan Antarmuka Halaman Utama



Gambar 3.11 Perancangan Antarmuka Halaman Utama

Penjelasan antarmuka halaman utama pada Gambar 3.11 dijelaskan secara detail pada Tabel 3.5 sebagai berikut:

Tabel 3.5 Perancangan Antarmuka Halaman Utama

Jenis	Nama	Keterangan
<i>Listview</i>	<i>List</i>	Untuk menampilkan hasil data pengingat yang telah dibuat
<i>Button</i>	Tambah	Jika <i>button</i> ditekan maka akan menampilkan menu tambah pengingat
<i>Button</i>	Map(<i>Icon</i>)	Jika <i>button</i> ditekan maka akan menampilkan lokasi yang pengguna tandai

3.6.2 Perancangan Antarmuka Tambah Peningat



Gambar 3.12 Perancangan Antarmuka Tambah Peningat

Penjelasan antarmuka tambah pengingat pada Gambar 3.12 dijelaskan secara detail pada Tabel 3.6 sebagai berikut:

Tabel 3.6 Perancangan Antarmuka Tambah Peningat

Jenis	Nama	Keterangan
<i>TextField</i>	Judul Peningat	Untuk memberikan judul pengingat
<i>TextField</i>	Deskripsi	Untuk memberikan deskripsi pada pengingat
<i>DateField</i>	Tanggal	Menampilkan tanggal
<i>TimeField</i>	Jam	Menampilkan jam
<i>Button</i>	Lokasi	Menampilkan maps untuk memilih lokasi
<i>Button</i>	Tambah Peningat	Untuk menambah data tambah pengingat

3.6.3 Perancangan Antarmuka Ubah Pengingat



Gambar 3.13 Perancangan Antarmuka Ubah Pengingat

Penjelasan antarmuka ubah pengingat pada Gambar 3.13 dijelaskan secara detail pada Tabel 3.7 sebagai berikut:

Tabel 3.7 Perancangan Antarmuka Ubah Pengingat

Jenis	Nama	Keterangan
<i>TextField</i>	Judul Pengingat	Untuk memberikan judul pada pengingat
<i>TextField</i>	Deskripsi	Untuk memberikan deskripsi pada pengingat
<i>DateField</i>	Tanggal	Jika <i>field</i> ditekan maka akan menampilkan tanggal kapan pengingat diinginkan pengingat
<i>TimeField</i>	Jam	Jika <i>field</i> ditekan maka akan menampilkan jam kapan pengingat diinginkan pengingat
<i>Button</i>	Lokasi	Jika diklik maka akan menampilkan <i>Maps</i> untuk memilih lokasi
<i>Button</i>	Ubah Pengingat	Untuk menyimpan perubahan data ubah pengingat

3.6.4 Perancangan Antarmuka Hapus Pengingat



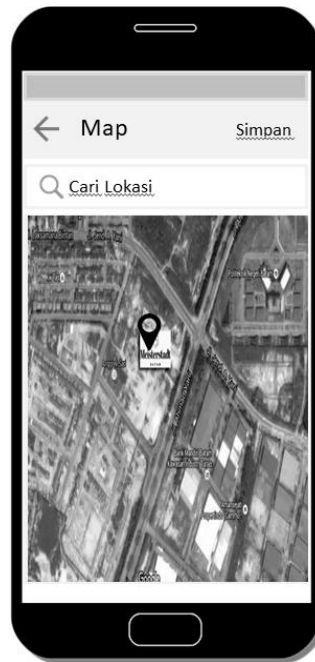
Gambar 3.15 Perancangan Antarmuka Hapus Pengingat

Penjelasan antarmuka hapus pengingat pada Gambar 3.15 dijelaskan secara detail pada Tabel 3.8 sebagai berikut:

Tabel 3.8 Perancangan Antarmuka Hapus Pengingat

Jenis	Nama	Keterangan
<i>Button</i>	Hapus (<i>Icon</i>)	Jika diklik maka akan menampilkan peringatan akan menghapus data
<i>AlertController</i>	Peringatan	Menampilkan konfirmasi sebelum menghapus data

3.6.5 Perancangan Antarmuka Lihat Lokasi Peningat



Gambar 3.16 Perancangan Antarmuka Lihat Lokasi Peningat

Penjelasan antarmuka lihat lokasi peringatan pada Gambar 3.16 dijelaskan secara detail pada Tabel 3.9 sebagai berikut:

Tabel 3.9 Perancangan Antarmuka Lihat Lokasi Peningat

Jenis	Nama	Keterangan
<i>MapView</i>	<i>Map</i>	Menampilkan map
<i>TextField</i>	Cari Lokasi	Untuk mencari lokasi yang ingin di tuju
<i>Button</i>	Simpan	Untuk menyimpan lokasi yang telah dipilih pengguna

BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi *Database*

Pada penelitian ini, penulis menggunakan database SQLite *Native Ionic*. *Database* hanya memiliki 1 tabel yaitu tabel pengingat. *Field* dari tabel pengingat disajikan pada tabel 4.1 sebagai berikut:

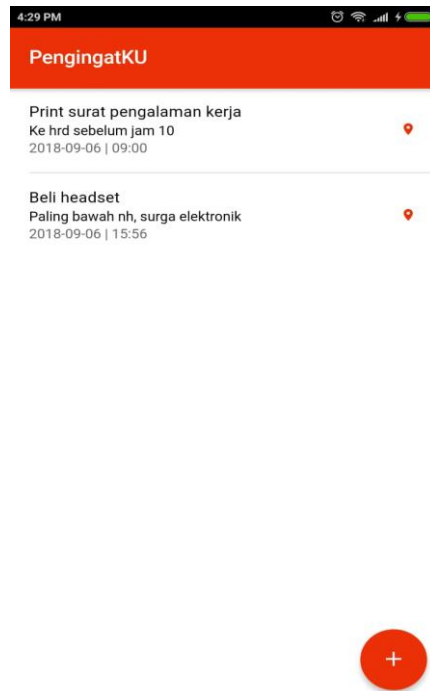
Tabel 4.1 Tabel Pengingat

Nama <i>Field</i>	Tipe Data
<i>rowid</i>	<i>INTEGER</i>
<i>title</i>	<i>TEXT</i>
<i>description</i>	<i>TEXT</i>
<i>date</i>	<i>TEXT</i>
<i>time</i>	<i>TEXT</i>
<i>lat</i>	<i>TEXT</i>
<i>lng</i>	<i>TEXT</i>
<i>jangkau</i>	<i>TEXT</i>
<i>notifid</i>	<i>TEXT</i>

4.2 Implementasi Antarmuka

4.2.1 Implementasi Antarmuka Halaman Utama

Halaman utama adalah halaman pertama ketika pengguna untuk pertama kalinya mengakses aplikasi ini. Pada halaman utama ini terdapat *listview* yang menampilkan data-data pengingat yang telah di kelola oleh pengguna. Antarmuka halaman utama disajikan pada Gambar 4.1.



Gambar 4.1 Implementasi Antarmuka Halaman Utama

Source code dari halaman utama pada aplikasi ini disajikan pada Gambar 4.2.

```

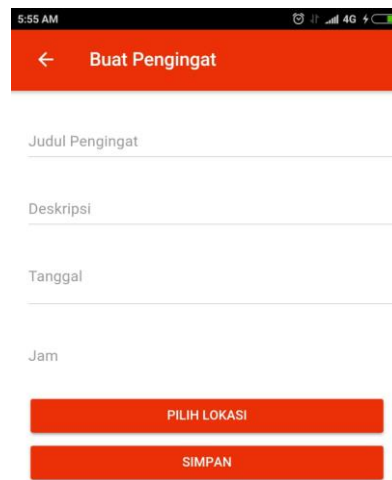
getData() {
  this.sqlite.create({
    name: 'ionidb.db',
    location: 'default'
  }).then((db: SQLiteObject) => {
    db.executeSql('CREATE TABLE IF NOT EXISTS pengingat(rowid INTEGER PRIMARY KEY, title TEXT,description TEXT, date TEXT, time TEXT, lat TEXT, lng TEXT, jangkau TEXT)', [])
    .then(res => console.log('Executed SQL'))
    .catch(e => console.log(e));
    db.executeSql('SELECT * FROM pengingat ORDER BY rowid DESC', [])
    .then(res => {
      this.pengingats = [];
      for(var i=0; i<res.rows.length; i++) {
        this.pengingats.push({rowid:res.rows.item(i).rowid, title:res.rows.item(i).title, description:res.rows.item(i).description,
        date:res.rows.item(i).date, time:res.rows.item(i).time, lat:res.rows.item(i).lat, lng:res.rows.item(i).lng, jangkau:res.rows.item(i)
        .jangkau})
      }
    })
  }).catch(e => console.log(e));
}

```

Gambar 4.2 Source Code Halaman Utama

4.2.2 Implementasi Antarmuka Halaman Tambah Pengingat

Halaman tambah pengingat adalah halaman yang akan tampil jika pengguna memilih tombol tambah pada halaman utama. Pada halaman tambah pengingat ini terdapat *form* untuk memasukkan data-data pengingat. Antarmuka halaman tambah pengingat disajikan pada Gambar 4.3.



The screenshot shows a mobile application interface for creating a reminder. At the top, there is a red header bar with a white back arrow on the left and the text 'Buat Pengingat' in white. Below the header, there are four text input fields with light gray placeholder text: 'Judul Pengingat', 'Deskripsi', 'Tanggal', and 'Jam'. At the bottom of the form, there are two red buttons with white text: 'PILIH LOKASI' and 'SIMPAN'. The background of the form is white, and the overall interface is clean and modern.

Gambar 4.3 Implementasi Antarmuka Halaman Tambah Pengingat

Source code dari halaman tambah pengingat disajikan pada gambar 4.4.

```

submit() {
  if( this.data.title && this.data.description && this.data.date && this.data.time && this.geoFenceProvider.lat && this.geoFenceProvider.lng) {
    var date = new Date(this.data.date + " " + this.data.time);
    var randNum = Math.floor(Math.random() * 1000);
    this.localNotifications.schedule([
      {
        id: randNum,
        text: this.data.title,
        trigger: { at: date },
        led: "FF0000",
        data: { title: this.data.title, desc: this.data.description, date: this.data.date },
        sound: this.setSound()
      }
    ]);
    this.sqlite.create({
      name: 'ionicdb.db',
      location: 'default'
    }).then((db: SQLiteObject) => {
      db.executeSql('INSERT INTO pengingat VALUES(NULL,?,?,?,?);',[this.data.title,this.data.description,this.data.date,this.data.time,
      this.geoFenceProvider.lat,this.geoFenceProvider.lng, 'false'])
        .then(res => {
          console.log(res);
          this.toast.show('Berhasil', '5000', 'center').subscribe(
            toast => {
              this.navCtrl.popToRoot();

```

```

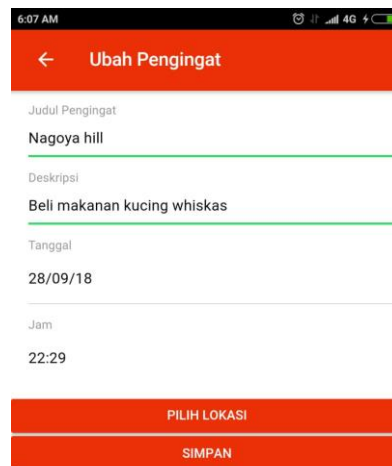
              this.navCtrl.popToRoot();
            }
          );
        });
      }.catch(e => {
        console.log(e);
        this.toast.show(e, '5000', 'center').subscribe(
          toast => {
            console.log(toast);
          }
        );
      });
    }).catch(e => {
      console.log(e);
      this.toast.show(e, '5000', 'center').subscribe(
        toast => {
          console.log(toast);
        }
      );
    });
  } else {
    let alert = this.alertCtrl.create({
      title: "Gagal!",
      subTitle: "Harap lengkapi semua data",
      buttons: ["OK"]
    });
    alert.present();
  }
}
}

```

Gambar 4.4 Source Code Halaman Tambah Pengingat

4.2.3 Implementasi Antarmuka Halaman Ubah Peningat

Halaman ubah pengingat adalah halaman yang akan tampil saat pengguna memilih tombol ubah pada data pengingat. Halaman ini memiliki tampilan yang mirip dengan halaman tambah pengingat, namun pada halaman ini fungsi yang berjalan pada *database* adalah fitur *update*. Antarmuka halaman ubah pengingat disajikan pada Gambar 4.5.



Gambar 4.5 Implementasi Antarmuka Halaman Ubah Peningat

Source code dari halaman ubah pengingat pada aplikasi ini disajikan pada Gambar 4.6.

```

this.sqlite.create({
  name: 'ionicdb.db',
  location: 'default'
}).then((db: SQLiteObject) => {
  db.executeSql('UPDATE pengingat SET title=?,description=?,date=?,time=?,lat=?,lng=? WHERE rowid=?',[this.data.title,this.data.description,
  this.data.date,this.data.time,this.geofenceProvider.lat,this.geofenceProvider.lng,this.data.rowid])
  .then(res => {
    console.log(res);
    this.toast.show('Data updated', '5000', 'center').subscribe(
      toast => {
        this.navCtrl.popToRoot();
      }
    );
  })
  .catch(e => {
    console.log(e);
    this.toast.show(e, '5000', 'center').subscribe(
      toast => {
        console.log(toast);
      }
    );
  });
}).catch(e => {
  console.log(e);
  this.toast.show(e, '5000', 'center').subscribe(
    toast => {
      console.log(toast);
    }
  );
});

let alert = this.alertCtrl.create({
  title: "Berhasil!",
  subTitle: "Data telah berhasil ditambah " + date,
  buttons: ["OK"]
});

alert.present();
}

```

Gambar 4.6 *Source Code* Halaman Ubah Pengingat

4.2.4 Implementasi Antarmuka Halaman Hapus Peningat

Halaman hapus pengingat adalah halaman utama yang akan tampil saat pengguna memilih tombol hapus. Fungsi yang berjalan pada halaman ini adalah fungsi *delete* pada *database*. Antarmuka halaman hapus pengingat disajikan pada Gambar 4.7.



Gambar 4.7 Implementasi Antarmuka Halaman Hapus Peningat

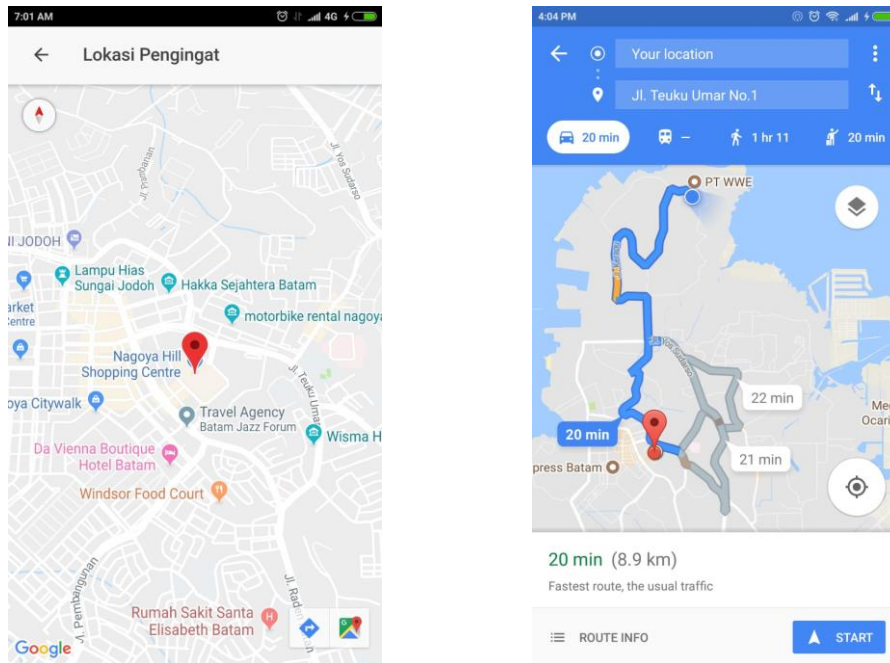
Source code dari halaman hapus pengingat pada aplikasi ini disajikan pada Gambar 4.8.

```
deleteData(rowid) {
  this.sqlite.create({
    name: 'ionicdb.db',
    location: 'default'
  }).then((db: SQLiteObject) => {
    db.executeSql('DELETE FROM pengingat WHERE rowid=?', [rowid])
      .then(res => {
        console.log(res);
        this.getData();
      })
      .catch(e => console.log(e));
  }).catch(e => console.log(e));
}
```

Gambar 4.8 *Source Code* Halaman Hapus Peningat

4.2.5 Implementasi Antarmuka Halaman Lihat Lokasi Peningat

Halaman lihat lokasi peringatan adalah halaman yang akan tampil saat pengguna memilih tombol map pada halaman utama seperti yang disajikan pada Gambar 4.9.



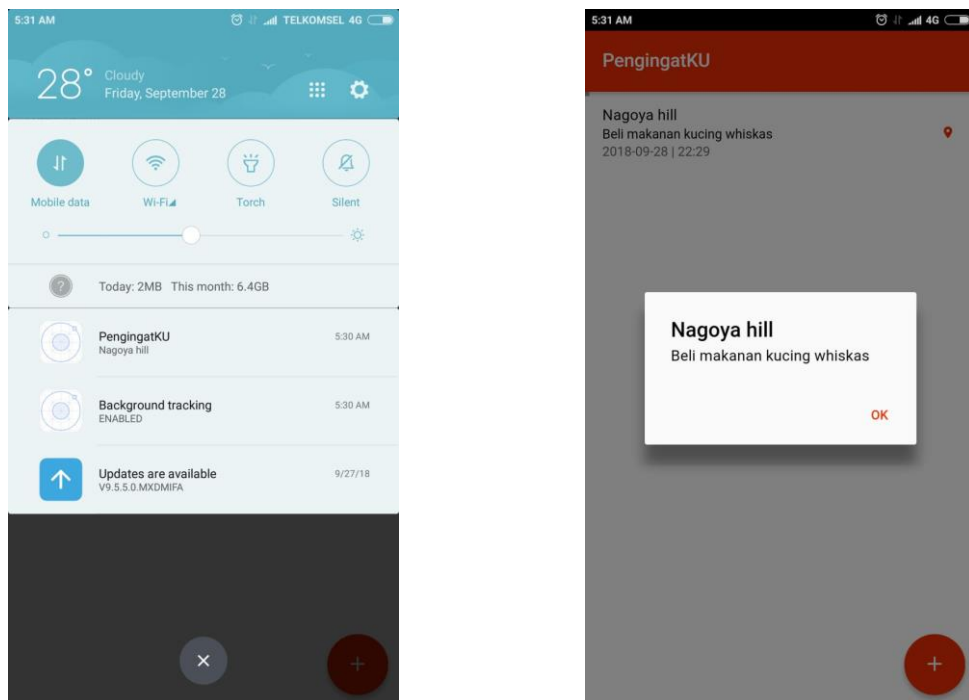
Gambar 4.9 Implementasi Antarmuka Lihat Lokasi Peningat
Source code dari halaman lihat lokasi pada aplikasi ini disajikan pada Gambar 4.10.

```
ionViewDidLoad() {  
  var dataFromHomePage = this.navParams.data;  
  this.lat = dataFromHomePage.lat;  
  this.lng = dataFromHomePage.lng;  
  this.loadMap();  
}  
loadMap() {  
  let mapOptions: GoogleMapOptions = {  
    camera: {  
      target: {  
        lat: this.lat,  
        lng: this.lng  
      },  
      zoom: 15,  
      tilt: 20  
    }  
  };  
  this.map = GoogleMaps.create("map", mapOptions);  
  this.map.addMarkerSync({  
    icon: "red",  
    animation: "BOUNCE",  
    position: {  
      lat: this.lat,  
      lng: this.lng  
    }  
  })  
}
```

Gambar 4.10 *Source Code* Lihat Lokasi Peningat

4.2.6 Implementasi Notifikasi

Pada penelitian ini sistem memberikan notifikasi kepada pengguna yaitu berupa pengingat yang akan ditampilkan jika waktu pada *database* sudah cocok dengan waktu yang dimasukkan oleh pengguna, dan jika pengguna berada dibawah dari 1 km berada di dekat lokasi tujuan. Antarmuka dari implementasi notifikasi terlihat seperti pada Gambar 4.11.



Gambar 4.11 Implementasi Notifikasi

4.3 Pengujian

Pengujian yang dilakukan pada aplikasi Peningat Aktivitas Berdasarkan Lokasi berbasis Android menggunakan pengujian *Black Box*. Metode pengujian *Black Box* adalah metode pengujian yang berdasarkan pada detail aplikasi, seperti fungsi-fungsi yang terdapat pada aplikasi, tampilan, dan proses sesuai alur fungsi yang telah dirancang. Terdapat beberapa fungsionalitas yang akan diuji, antara lain:

1. Tambah Peningat

2. Ubah Pengingat
3. Hapus Pengingat
4. Lihat Lokasi
5. Notifikasi Pengingat

Hasil pengujian menunjukkan bahwa Aplikasi Pengingat Aktivitas Berdasarkan Lokasi berbasis Android ini berjalan sesuai dengan yang telah dirancang sebelumnya. Detail dari pengujian aplikasi dapat dilihat pada Lampiran A.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis, perancangan, implementasi dan pengujian yang telah dilakukan, maka dapat disimpulkan:

1. Aplikasi dapat diimplementasikan sesuai dengan rancangan yang telah dibuat.
2. Aplikasi dapat melakukan tambah pengingat, ubah pengingat, dan hapus pengingat.
3. Aplikasi dapat menggunakan *maps*.
4. Aplikasi dapat melihat lokasi pada *maps* dan juga memberikan direksi menuju lokasi tujuan.
5. Aplikasi dapat memberikan notifikasi pengingat kepada pengguna.

5.2 Saran

Mengingat aplikasi Pengingat aktivitas masih memiliki banyak kekurangan, maka penulis memberikan beberapa saran yang diharapkan dapat digunakan untuk pengembangan dari aplikasi ini selanjutnya. Saran-saran tersebut antara lain:

1. Aplikasi dapat menyimpan tempat favorit berdasarkan lokasi yang sering dipilih oleh pengguna.
2. Data pengingat secara langsung dapat tersinkronisasi dengan kalender yang ada di *smartphone* pengguna.
3. Fitur pengingat berdasarkan lokasi dapat berjalan tanpa harus membuka aplikasi terlebih dahulu.

Daftar Pustaka

A., Fitrullah, dan Ardiansyah, A., 2016. Layanan Informasi Lokasi Event Menggunakan Lbs Berbasis JQuery Mobile. Jurnal Informatika, 10(2).

Ardhana, K.M. Yosef., 2014. Pemrograman Android Black Box. Jakarta:Jasakom.

Djamarah, Syaiful Bahri. 2008, Psikologi Belajar, Jakarta: Rineka Cipta.

Iqbal, M., Husni, M. and Studiawan, H., 2012. Implementasi Klien SIP Berbasis Web Menggunakan HTML5 dan Node. js. Jurnal Teknik ITS, 1(1), pp.A242-A245.

Imaniar, J. and Arifin, A.S.K., 2011. Aplikasi Location Based Service untuk Sistem Informasi Publikasi Acara pada Platform Android. Skripsi S-1, ITS, Surabaya.

Ionic Framework. URI = <http://ionicframework.com/docs/guide/preface.html>.

Kadir, Abdul., 2012. Algoritma dan Pemrograman Menggunakan Java. Yogyakarta:Penerbit Andi

Kusuma, Ayu. 2013. Aplikasi pencarian Apotek 24 jam di Kota Semarang. Semarang.

Komputer, Wanaha., 2009.Menguasai Pemrograman Web dengan JavaScript 2009. Semarang: Penerbit Andi.

L, Neelu., Aruna, Kumara.B., Shashidhar.V., Bharath J., 2015. "Location Based Reminder Using Android and *Google maps*". International Journal of

Innovative Research in Computer and Communication Engineering. Vol. 3, Issue 5, May 2015

Mahdia, F. and Noviyanto, F., 2013. Pemanfaatan *Google maps API* Untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web (Studi Kasus: Badan Penanggulangan Bencana Daerah Kota Yogyakarta). *Jurnal Sarjana Teknik Informatika*, 1(1).

Perakovic, Dragan., Vladimir Remenar., “Reminder Based On The User’s Location”. University of Zagreb Faculty of Transport and Traffic Sciences

Safaat, H. Nazaruddin., 2012. *Android*. Bandung: Informatika Bandung.

SETIYAD, A. dan Harihayati, T., 2015. PENERAPAN SQLITE PADA APLIKASI PENGATURAN WAKTU UJIAN DAN PRESENTASI. *Majalah Ilmiah UNIKOM*, 13(2).

Shah, Priyanka., Ruta, Gadgil., Neha, Tamhankar., 2012. "Location Based Reminder Using GPS for Mobile (Android)". *ARPN Journal of Science and Technology*. VOL. 2, NO. 4, May 2012

Sianipar, R.H., 2015. *HTML5 & CSS3*. Bandung: Informatika

Sohn, Timothy.dkk., “Place-Its: A Study of Location-Based Reminders on Mobile Phones”. Computer Science and Engineering University of California, San Diego, La Jolla, CA, USA

Sugiarto, I., 2013. *Aplikasi Pencarian Lokasi Terdekat Pelayanan Kesehatan Berbasis Android Di Yogyakarta*. Skripsi tidak dipublikasikan: STIMIK AMIKOM Yogyakarta.

Susanto, Azhar., 2017. Informasi Akuntansi dan Sistem Informasi Manajemen. Bandung: Lingga Jaya

Yulianto,dkk ., 2018. Penerapan Formula Haversine pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal. Jurnal Ilmiah Ilmu Komputer. Universitas Mulawarman Samarinda Kalimantan Timur.