



**INSTITUT TEKNOLOGI DEL**

**Pengembangan Aplikasi *Track My Trip* Menerapkan Konsep  
CICD (*Continuous Integration Continuous Delivery*)**

**TUGAS AKHIR**

**11317012    Jhon Mejer Panjaitan**

**11317028    Boy S A Hutagaol**

**11317029    Delta Pangaribuan**

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO  
PROGRAM STUDI DIPLOMA 3 TEKNOLOGI INFORMASI**

**LAGUBOTI  
AGUSTUS 2020**



**INSTITUT TEKNOLOGI DEL**

**Pengembangan Aplikasi *Track My Trip* Menerapkan Konsep  
CICD (*Continuous Integration Continuous Delivery*)**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar  
Diploma 3**

**Program Studi Teknologi Informasi**

**11317012 Jhon Mejer Panjaitan**

**11317028 Boy S A Hutagaol**



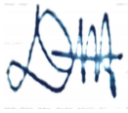
**11317029 Delta Pangaribuan**

**FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO  
PROGRAM STUDI DIPLOMA 3 TEKNOLOGI INFORMASI**

**LAGUBOTI  
AGUSTUS 2020**

## HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

1. NAMA : JHON MEJER PANJAITAN  
NIM : 11317012  
TANDA TANGAN : 
  2. NAMA : BOY S A HUTAGAOL  
NIM : 11317028  
TANDA TANGAN : 
  3. NAMA : DELTA PANGARIBUAN  
NIM : 11317029  
TANDA TANGAN : 
- TANGGAL : 3 Agustus 2020

## HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

1. Nama : Jhon Meyer Panjaitan  
NIM : 11317012  
Program studi : Diploma 3 Teknologi Informasi
2. Nama : Boy S A Hutagaol  
NIM : 11317028  
Program studi : Diploma 3 Teknologi Informasi
3. Nama : Delta Pangaribuan  
NIM : 11317029  
Program studi : Diploma 3 Teknologi Informasi  
Judul Tugas Akhir : **Pengembangan Aplikasi Track My Trip Menerapkan Konsep CICD**  
(*Continuous Integration Continuous Delivery*)

Telah berhasil dipertahankan dihadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Diploma 3 program studi Diploma 3 Teknologi Informasi Fakultas Informatika dan Teknik Elektro Institut Teknologi Del.

## DEWAN PENGUJI

- Pembimbing : Mario Elyezer Subekti Simaremare ( )
- Pembimbing : Teamsar Muliadi Panggabean ( )
- Penguji : Roy Dedy Hasiholan Lumbantobing ( )
- Penguji : Eka Stephani Sinambela ( )

Ditetapkan : Laguboti  
Tanggal : 3 Agustus 2020

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan kasih karunia-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik dan sesuai dengan waktu yang telah ditentukan. Penulisan Tugas Akhir ini disusun sebagai persyaratan kelulusan Program Studi Diploma 3 Teknologi Informasi di Institut Teknologi Del. Adapun yang menjadi judul dari Tugas Akhir ini adalah **“Pengembangan Aplikasi Track My Trip dengan Menerapkan konsep CICD (*Continuous Integration Continuous Delivery*)”**. Tujuan dari pengerjaan Tugas Akhir ini adalah untuk mengembangkan aplikasi berbasis android yang digunakan untuk melakukan perekaman perjalanan secara langsung, merelasikan multimedia seperti foto, video, note dengan titik koordinat lokasi pengambilan multimedia dengan memanfaatkan GPS (*Global Positioning System*) dan sensor yang terdapat dalam *smartphone* atau handset. Selama pengerjaan dan penulisan dokumen Tugas Akhir ini, semua tidak terlepas dari dukungan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada orang tua, pembimbing, penguji, rekan-rekan kelompok dan sivitas Institut Teknologi Del dalam memberikan bimbingan, arahan, motivasi, doa dan dukungan dalam proses penyelesaian Tugas Akhir ini.

Dalam penulisan Tugas Akhir ini, penulis menyadari dalam Tugas Akhir terdapat banyak kesalahan dan kekurangan, penulis memohon maaf atas segala kekurangan dan kekurangan penulisan tersebut dan menerima saran dan kritik yang membangun kepada penulis demi pengembangan Tugas Akhir ini agar lebih baik lagi. Akhir kata, semoga Tugas Akhir ini berguna dan bermanfaat bagi pembaca.

Sitoluama, 3 Agustus 2020

Penulis

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI DOKUMEN TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Institut Teknologi Del, saya yang bertanda tangan di bawah ini:

1. Nama : Jhon Mejer Panjaitan  
NIM : 11317012  
Fakultas/Program Studi : Fakultas Informatika dan Teknik Elektro  
Program Studi Diploma 3 Teknologi Informasi
2. Nama : Boy S A Hutagaol  
NIM : 11317028  
Fakultas/Program Studi : Fakultas Informatika dan Teknik Elektro  
Program Studi Diploma 3 Teknologi Informasi
3. Nama : Delta Pangaribuan  
NIM : 11317029  
Fakultas/Program Studi : Fakultas Informatika dan Teknik Elektro  
Program Studi Diploma 3 Teknologi Informasi  
Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Del **Hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul: **Pengembangan Aplikasi Track My Trip Menerapkan Konsep CICD (*Continuous Integration Continuous Delivery*)**. Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-eksklusif ini Institut Teknologi Del berhak menyimpan, mengalih/media-format dalam bentuk pangkalan data (database), merawat dan mempublikasikan Tugas Akhir kami selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.  
Demikian pernyataan ini kami buat dengan sebenarnya.

Dibuat di : Sitoluama

Pada tanggal : 3 Agustus 2020

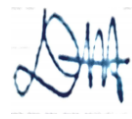
Yang menyatakan,



(Jhon Mejer Panjaitan)



(Boy S A Hutagaol)



(Delta Pangaribuan)

## ABSTRAK

Nama : D3TI-13  
Program Studi : Diploma 3 Teknologi Informasi  
Judul : Pengembangan Aplikasi Track My Trip Menerapkan Konsep CICD (*Continuous Integration Continuous Delivery*)

Melakukan dokumentasi perjalanan merupakan salah satu cara yang sering dilakukan seseorang untuk menyimpan data perjalanan yang telah dilalui. Hal ini bertujuan agar suatu saat data dokumentasi perjalanan dapat dibuka, dilihat dan bahkan dibagikan dengan orang lain yang ingin mengetahui perjalanan yang telah dilalui. Dengan demikian orang lain dapat mengetahui gambaran tentang lokasi atau perjalanan yang telah dilakukan oleh orang yang melakukan perjalanan tersebut.

Untuk melakukan hal tersebut maka dikembangkan aplikasi Track My Trip dengan tujuan aplikasi dapat digunakan oleh masyarakat umum untuk mendokumentasikan perjalanan. Dengan adanya aplikasi Track My Trip pengguna akan mudah menyimpan dokumentasi perjalanan yang dilakukan, karena aplikasi ini dapat digunakan untuk menyimpan koordinat lokasi yang dilalui, dimana saat melakukan perhentian dilokasi tertentu pengguna aplikasi dapat mengasosiasikan foto, note, suara dan video dengan koordinat tempat pengambilan multimedia tersebut. Dengan demikian data dokumentasi perjalanan tersimpan. Pada pengembangan aplikasi ini akan menggunakan menerapkan konsep *automation* pada saat implementasi yaitu CICD (*Continuous Integration Continuous Delivery*). CICD adalah konsep *automation* yang akan memberikan kemudahan merelasikan kode, proses pembuatan dan pemantauan yang sederhana serta melakukan *deploy* di berbagai lingkungan yang diperlukan menggunakan *automation tools* untuk meningkatkan efisiensi dan efektivitas pengembangan aplikasi pada saat implementasi.

**Keywords:** CICD, Perekaman perjalanan, *Tracking*

## ABSTRACT

Name : D3TI-13  
Study Program : Diploma 3 Teknologi Informasi  
Title : Pengembangan Aplikasi Track My Trip Menerapkan Konsep CICD (*Continuous Integration Continuous Delivery*)

*Doing travel documentation is one way that is often done by someone to save data on a trip that has been traversed. This aims to someday travel documentation data can be opened, seen and even shared with others who want to know the journey that has been passed. Thus, other people can find out about the location or the trip that has been done by the person who made the trip.*

*To do this, the Track My Trip application was developed with the aim that the application can be used by the general public to document the trip. With the track my trip application, people will easily save the documentation of the trip, because this application can be used to store the coordinates of the location being traversed, where when stopping at a certain location the application user can associate photos, notes, sound and video with the coordinates of the place where the multimedia was taken. Thus, the travel documentation data is stored. In developing this application, it will use the concept of automation at the time of implementation, namely CICD (*Continuous Integration Continuous Delivery*). CICD is an automation concept that will make it easier to correlate code, a simple manufacturing ad monitoring process and deploy in various environments as needed using automation tools to improve the efficiency and effectiveness of application development during implementation.*

**Keywords:** CICD, Perekaman perjalanan, Tracking



## DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS .....	ii
HALAMAN PENGESAHAN .....	iii
KATA PENGANTAR .....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI DOKUMEN TUGAS	
AKHIR UNTUK KEPENTINGAN AKADEMIS.....	v
ABSTRAK .....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR.....	xv
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan.....	4
1.4 Lingkup.....	4
1.5 Pendekatan .....	4
1.6 Sistematika Penyajian.....	6
BAB 2 TINJAUAN PUSTAKA .....	7
2.1 <i>Tracking</i> .....	7
2.2 GPS.....	7
2.3 Sensor .....	9
2.4 Android .....	10
2.4.1 Java Programming Language.....	13
2.4.2 Firebase .....	13
2.5 Google Maps API.....	14
2.6 <i>CICD (Continuous Integration Continuous Delivery)</i> .....	14
2.6.1 Jenkins .....	16
2.6.2 Instalasi Jenkins dan Konfigurasi <i>Repository</i> dengan Jenkins .....	17
2.7 Kesimpulan .....	23
BAB 3 ANALISIS DAN DESAIN .....	24
3.1 Analisis Navigasi <i>Tracking</i> .....	24
3.2 Analisis Aplikasi <i>Tracking</i> Sejenis.....	25
3.2.1 PolarSteps .....	25
3.2.2 Rute Saya.....	26

3.2.3	Google Maps.....	27
3.2.4	Maverick.....	28
3.2.5	Jejak Kaki .....	29
3.2.6	Waze .....	30
3.2.7	Here WeGo.....	31
3.2.8	A-GPS Tracker .....	32
3.3	Analisis GPS ( <i>Global Positioning System</i> ) .....	36
3.4	Analisis Penggunaan Multimedia .....	40
3.5	Analisis CICD dengan <i>Tools</i> Jenkins .....	41
3.6	Desain Arsitektur Aplikasi.....	42
3.7	Analisis Kebutuhan.....	43
3.7.1	<i>Functional Requirement</i> .....	43
3.7.2	<i>Business Process</i> .....	44
3.7.2.1	Melakukan Autentikasi.....	44
3.7.2.2	Menunjukkan Lokasi Pengguna .....	45
3.7.2.3	Rekam Perjalanan .....	46
3.7.2.4	Menghapus Data Perjalanan.....	47
3.7.2.5	Merelasikan Lokasi dengan Foto.....	47
3.7.2.6	Merelasikan Lokasi dengan Video .....	48
3.7.2.7	Merelasikan Lokasi dengan Note .....	49
3.7.2.8	Merelasikan Lokasi dengan voice record.....	49
3.7.2.9	Share Perjalanan .....	50
3.7.2.10	Impor Data Perjalanan.....	50
3.7.3	Non-Functional Requirement.....	51
3.8	Desain.....	51
3.8.1	<i>User Characteristic</i> .....	52
3.8.1.1	<i>User Group 1</i> .....	52
3.8.2	Rancangan Aplikasi Track My Trip.....	52
3.8.2.1	Use Case Diagram .....	52
3.8.2.2	Use Case Scenario .....	55
3.8.2.2.1	Use Case Melakukan Autentikasi.....	55
3.8.2.2.2	Use Case Scenario Menunjukkan Lokasi Pengguna .....	57
3.8.2.2.3	Use Case Scenario Merekam Perjalanan .....	58
3.8.2.2.4	Use Case Scenario Menghapus Data Perjalanan.....	59
3.8.2.2.5	Use Case Scenario Merelasikan Lokasi dengan Foto .....	61

3.8.2.2.6	Use Case Scenario Merelasikan Lokasi dengan Video .....	62
3.8.2.2.7	Use Case Scenario Merelasikan Lokasi dengan Video .....	63
3.8.2.2.8	Use Case Merelasikan Lokasi dengan Voice Record .....	65
3.8.2.2.9	Use Case Scenario Share perjalanan .....	66
3.8.2.2.10	Use Case Scenario Impor Data Perjalanan .....	67
3.8.3	Rancangan Kebutuhan Data .....	69
3.8.3.1	Conceptual Data Model.....	71
3.8.3.2	Physical Data Model .....	72
3.8.3.3	Class Diagram .....	73
3.8.3.4	Sequence Diagram.....	74
3.8.3.4.1	Melakukan Autentikasi .....	74
3.8.3.4.2	Menunjukkan Lokasi Pengguna.....	75
3.8.3.4.3	Rekam Perjalanan.....	75
3.8.3.4.4	Menghapus Data Perjalanan .....	76
3.8.3.4.5	Merelasikan Lokasi dengan Foto .....	77
3.8.3.4.6	Merelasikan Lokasi dengan Video .....	78
3.8.3.4.7	Merelasikan Lokasi dengan Note .....	79
3.8.3.4.8	Merelasikan Lokasi dengan Voice Record.....	80
3.8.3.4.9	Share Perjalanan.....	81
3.8.3.4.10	Impor Data Perjalanan.....	82
3.8.3.5	Desain <i>User Interface</i> Aplikasi .....	82
3.8.3.5.1	Desain <i>User Interface</i> Home .....	83
3.8.3.5.2	Desain <i>User Interface</i> Register.....	84
3.8.3.5.3	Desain <i>User Interface</i> Login.....	84
3.8.3.5.4	Desain <i>User Interface</i> Lokasi dari Pengguna.....	85
3.8.3.5.5	Desain <i>User Interface</i> Rekam Perjalanan .....	86
3.8.3.5.6	Desain <i>User Interface</i> Menghapus Data Perjalanan.....	87
3.8.3.5.7	Desain <i>User Interface</i> Merelasikan Lokasi dengan Foto.....	88
3.8.3.5.8	Desain <i>User Interface</i> Merelasikan Lokasi dengan Note .....	89
3.8.3.5.9	Desain <i>User Interface</i> Merelasikan Lokasi dengan Video.....	90
3.8.3.5.10	Desain <i>User Interface</i> Merelasikan Lokasi dengan Voice .....	91
3.8.3.5.11	Desain <i>User Interface</i> About.....	92
3.8.3.5.12	Desain <i>User Interface</i> Share .....	93
3.8.3.5.13	Desain <i>User Interface</i> Impor .....	94
BAB 4	IMPLEMENTASI.....	96

4.1	Kebutuhan Implementasi .....	96
4.2	Batasan Implementasi.....	97
4.3	Persiapan Praktik <i>Continuous Integration Continuous Delivery</i> .....	97
4.4	Proses Implementasi Aplikasi Track My Trip .....	98
4.4.1	Implementasi Fungsi pada Aplikasi Track My Trip .....	98
4.4.1.1	Implementasi Menu Home .....	98
4.4.1.2	Implementasi Fungsi Autentikasi .....	99
4.4.1.3	Implementasi Fungsi Lokasi Pengguna.....	100
4.4.1.4	Implementasi Fungsi Rekam Perjalanan .....	101
4.4.1.5	Implementasi Fungsi Merelasikan Lokasi dengan Foto .....	101
4.4.1.6	Implementasi Fungsi Merelasikan Lokasi dengan Video .....	102
4.4.1.7	Implementasi Fungsi Merelasikan Lokasi dengan Voice Record ..	102
4.4.1.8	Implementasi Fungsi Merelasikan Lokasi dengan Note.....	104
4.4.2	Implementasi Penerapan <i>Continuous Integration Continuous Delivery</i> ....	105
4.4.2.1	Pengelolaan <i>Repository</i> .....	105
4.4.2.2	Penerapan CICD dengan Jenkins .....	106
BAB 5 PENGUJIAN .....		111
5.1	Tujuan Pengujian .....	111
Tujuan dari pengujian terhadap aplikasi yang dibangun adalah sebagai berikut: ....		111
5.2	Skenario Pengujian Aplikasi .....	111
5.2.1	BU-01 Autentikasi.....	112
5.2.2	BU-02 Menunjukkan Lokasi Pengguna .....	113
5.2.3	BU-03 Rekam Perjalanan .....	114
5.2.4	BU-04 Menghapus Data Perjalanan.....	115
5.2.5	BU-05 Merelasikan Lokasi dengan Foto.....	116
5.2.6	BU-06 Merelasikan Lokasi dengan Video.....	116
5.2.7	BU-07 Merelasikan Lokasi dengan Voice Record .....	117
5.2.8	BU-08 Merelasikan Lokasi dengan Note .....	118
5.2.9	BU-09 Share Perjalanan .....	119
5.2.10	BU-10 Impor Data Perjalanan .....	120
BAB 6 HASIL DAN PEMBAHASAN .....		122
6.2	Hasil Aplikasi .....	122
6.2.1	Splashscreen .....	122
6.2.2	Home .....	123
6.2.3	Autentikasi .....	123

6.2.4	Lokasi Pengguna.....	125
6.2.5	Rekaman Perjalanan.....	125
6.2.6	Foto dengan Lokasi.....	126
6.2.7	Video dengan Lokasi.....	126
6.2.8	Voice record dengan Lokasi .....	127
6.2.9	Note dengan Lokasi .....	127
6.3	Pembahasan.....	128
BAB 7 KESIMPULAN DAN SARAN.....		130
7.1	Kesimpulan .....	130
7.2	Saran .....	130
Daftar Pustaka .....		131

## DAFTAR TABEL

Tabel 2.1. Versi Android .....	11
Tabel 3.1. Analisis Perbandingan Aplikasi Sejenis .....	34
Tabel 3.2. <i>Functional Requirement</i> .....	43
Tabel 3.3. Non-Functional Requirement .....	51
Tabel 3.4. Use Case Scenario Melakukan Autentikasi .....	55
Tabel 3.5. Use Case Scenario Menunjukkan Lokasi Pengguna.....	57
Tabel 3.6. Use Case Scenario Merekam Perjalanan.....	58
Tabel 3.7. Use Case Scenario Menghapus Data Perjalanan .....	60
Tabel 3.8. Use Case Scenario Merelasikan Lokasi dengan Foto .....	61
Tabel 3.9. Use Case Scenario Merelasikan Lokasi dengan Video .....	62
Tabel 3.10. Use Case Scenario Merelasikan Lokasi dengan Note .....	63
Tabel 3.11. Use Case Scenario Merelasikan Lokasi dengan Voice Record .....	65
Tabel 3.12. Use Case Scenario Share Perjalanan .....	66
Tabel 3.13. Use Case Scenario Impor Data Perjalanan.....	67
Tabel 4.1. Kebutuhan Implementasi pada Software.....	96
Tabel 4.2. Kebutuhan Implementasi pada Hardware.....	97
Tabel 4.3. Potongan Kode Program Implementasi Home.....	99
Tabel 4.4. Potongan Kode Program Implementasi Register .....	99
Tabel 4.5. Potongan Kode Program Implementasi Login .....	100
Tabel 4.6. Potongan Kode Program Implementasi Lokasi Pengguna.....	100
Tabel 4.7. Potongan Kode Program Implementasi Rekam Perjalanan .....	101
Tabel 4.8. Potongan Kode Program Implementasi Lokasi dengan Foto .....	101
Tabel 4.9. Potongan Kode Program Implementasi Lokasi dengan Video .....	102
Tabel 4.10. Potongan Kode Program Implementasi Lokasi dengan Voice Record .....	103
Tabel 4.11. Potongan Kode Program Implementasi Lokasi dengan Note .....	104
Tabel 4.12. Daftar <i>Build</i> .....	109
Tabel 5.1. Butir Uji Autentikasi.....	112
Tabel 5.2. Butir Uji Menunjukkan Lokasi Pengguna .....	113
Tabel 5.3. Butir Uji Rekam Perjalanan .....	114
Tabel 5.4. Butir Uji Menghapus Data Perjalanan.....	115
Tabel 5.5. Butir Uji Merelasikan Lokasi dengan Foto.....	116
Tabel 5.6. Butir Uji Merelasikan Lokasi dengan Video .....	117
Tabel 5.7. Butir Uji Merelasikan Lokasi dengan Voice Record .....	118

Tabel 5.8. Butir Uji Merelasikan Lokasi dengan Note .....	118
Tabel 5.9. Butir Uji <i>Share</i> Perjalanan.....	119
Tabel 5.10. Butir Uji Impor Data Perjalanan .....	120
Tabel 6.1. Perbandingan Pengembangan Perangkat Lunak dengan CICD dan Tanpa CICD .....	128

## **DAFTAR GAMBAR**

Gambar 2.1. Trilaterasi Dalam Global Positioning System (GPS)	9
Gambar 2.2. Arsitektur Sistem Operasi Android	12
Gambar 2.3. Arsitektur Sistem Firebase	14
Gambar 2.4. Arsitektur sebuah CICD	15
Gambar 2.5. Jenkins Job List and Configurations	16
Gambar 2.6. Webpage Running	17
Gambar 2.7. Customize Jenkins	18
Gambar 2.8. Create First Admin User	18
Gambar 2.9. Sign In	19
Gambar 2.10. Jenkins Dashboard	19
Gambar 2.11. Jenkins Environment Variables	20
Gambar 2.12. Jenkins Job Style	20
Gambar 2.13. Jenkins Credentials Provider	21
Gambar 2.14. Jenkins Source Code Management	21
Gambar 2.15. Process Build	22
Gambar 2.16. Success Build	22
Gambar 2.17. Jenkins Console Output	23
Gambar 3.1. Tampilan Polarstep	26
Gambar 3.2. Tampilan RuteSaya	27
Gambar 3.3. Tampilan GoogleMaps	28
Gambar 3.4. Tampilan Maverick	29
Gambar 3.5. Tampilan Jejak Kaki	30
Gambar 3.6. Tampilan Waze	31
Gambar 3.7. Tampilan HereWeGo	32
Gambar 3.8. Tampilan A-GPS Tracker	33
Gambar 3.9. Data Pengambilan GPS Menggunakan Device 1	37
Gambar 3.10. Data Pengambilan GPS Menggunakan Device 2	37
Gambar 3.11. Data Pengambilan GPS Lebih Kecil 10 Meter	38
Gambar 3.12. Data Pengambilan GPS Lebih Besar 10 Meter	38
Gambar 3.13. Ilustrasi Pengambilan Titik Koordinat	39
Gambar 3.14. Interaksi antara User, Device dan GPS	39
Gambar 3.15. Analisis Data Multimedia	40
Gambar 3.16. Analisis Proses CICD	42



Gambar 3.17. Arsitektur Aplikasi	42
Gambar 3.18. Business Process Melakukan Autentikasi	45
Gambar 3.19. Business Process Menampilkan Lokasi	46
Gambar 3.20. Business Process Rekam Perjalanan	46
Gambar 3.21. Business Process Menghapus Perjalanan	47
Gambar 3.22. Business Process Merelasikan Lokasi dengan Foto	48
Gambar 3.23. Business Process Merelasikan Lokasi dengan Video	48
Gambar 3.24. Business Process Merelasikan Lokasi dengan Note	49
Gambar 3.25. Business Process Merelasikan Lokasi dengan Voice	50
Gambar 3.26. Business Process Share perjalanan	50
Gambar 3.27. Business Process Impor Data Perjalanan	51
Gambar 3.28. <i>Use Case Diagram</i>	54
Gambar 3.29. Entity Relationship Diagram	70
Gambar 3.30. Conceptual Data Model	71
Gambar 3.31. Physical Data Model	72
Gambar 3.32. Class Diagram	73
Gambar 3.33. Sequence Diagram Melakukan Autentikasi	74
Gambar 3.34. Sequence Diagram Lokasi Pengguna	75
Gambar 3.35. Sequence Diagram Rekam Perjalanan	76
Gambar 3.36. Sequence Diagram Menghapus Data Perjalanan	77
Gambar 3.37. Sequence Diagram Merelasikan Lokasi dengan Foto	78
Gambar 3.38. Sequence Diagram Merelasikan Lokasi dengan Video	79
Gambar 3.39. Sequence Diagram Merelasikan Lokasi dengan Note	80
Gambar 3.40. Sequence Diagram Merelasikan Lokasi dengan Voice Record	81
Gambar 3.41. Sequence Diagram Share perjalanan	81
Gambar 3.42. Sequence Diagram Impor Data Perjalanan	82
Gambar 3.43. Design UI Home	83
Gambar 3.44. Desain UI Register	84
Gambar 3.45. Desain UI Login	85
Gambar 3.46. Design UI Lokasi Pengguna	86
Gambar 3.47. Design UI Perjalanan	87
Gambar 3.48. Desain UI Menghapus Data Perjalanan	88
Gambar 3.49. Design UI Menambahkan Foto	89
Gambar 3.50. Design UI Menambahkan Note	90
Gambar 3.51. Design UI Menambahkan Video	91

Gambar 3.52. Design UI untuk Menambahkan Audio	92
Gambar 3.53. Design UI About	93
Gambar 3.54. Design User Interface Share perjalanan	94
Gambar 3.55. Design User Interface Impor Data Perjalanan	95
Gambar 4.1. <i>Repository</i> pada Gitlab	105
Gambar 4.2. <i>Build Schedule</i>	106
Gambar 4.3. <i>Console Output</i>	107
Gambar 4.4. Build History	108
Gambar 4.5. <i>Email Notification</i>	108
Gambar 6.1. Tampilan Splashscreen	122
Gambar 6.2. Tampilan Home	123
Gambar 6.3. Tampilan Register	124
Gambar 6.4. Tampilan Login	124
Gambar 6.5. Tampilan Lokasi Pengguna	125
Gambar 6.6. Tampilan Track Journey	125
Gambar 6.7. Tampilan Ambil Gambar	126
Gambar 6.8. Tampilan Rekam Video	126
Gambar 6.9. Tampilan Record Suara	127
Gambar 6.10. Tampilan Note	127

## BAB 1

### PENDAHULUAN

Pendahuluan berisi deskripsi umum dari kajian yang dikerjakan. Deskripsi ini mencakup latar belakang mengapa topik ini dipilih, tujuan dari kajian yang dilakukan, dan sistematika penyajian laporan hasil kajian yang dilakukan.

#### 1.1 Latar Belakang

*Tracking* sangat dibutuhkan pada saat ini karena manfaat dari *tracking* yang dapat mempermudah kehidupan manusia dan mengatasi berbagai masalah [1]. *Tracking* secara harafiah memiliki arti mengikuti jalan atau kegiatan untuk mengikuti jejak suatu obyek. Pengertian *tracking* atau pemantauan dalam hal ini adalah kegiatan untuk memantau perjalanan (*travel*) seseorang berdasarkan posisi yang telah dilalui yang didapat dari *smartphone* [2]. Misalnya, orang tua yang ingin melacak keberadaan anaknya, ketika seseorang yang ingin mencari *smartphone* yang hilang, ketika seseorang yang ingin melacak perjalanannya, untuk mengetahui lokasi suatu tempat seperti sekolah, kantor, dll. Untuk melakukan *tracking* perjalanan, dibutuhkan suatu teknologi yaitu GPS (*Global Positioning System*) [3]. GPS adalah teknologi yang menggunakan satelit dan metode Triangulasi untuk menentukan navigasi dan posisi secara global [4].

Beberapa aplikasi yang menggunakan GPS diantaranya Maps, Maps.Me, Waze, Here WeGo, MapFactor, MapQuest, Polaris GPS Navigation, Scout GPS Navigation & Meet Up. Salah satu contoh fungsi GPS adalah pada aplikasi Maps yaitu untuk menentukan lokasi koordinat atau posisi suatu tempat yang tersebar di seluruh dunia. Contoh aplikasi navigasi yang sudah ada yaitu PolarSteps. Aplikasi PolarSteps secara otomatis melacak rute dan tempat yang dikunjungi selama perjalanan. Namun, aplikasi tidak dapat merelasikan data multimedia dengan koordinat yang dilalui.

Penelitian sebelumnya yaitu aplikasi perjalanan GPS *tracking* berbasis web pada handphone J2ME [5]. Pada penelitian ini menghasilkan aplikasi yang mampu menunjukkan lokasi dimana pengguna aplikasi berada dan mampu menyimpan *history* perjalanan pengguna lalu. Namun, pada aplikasi tidak dapat merelasikan

antara waktu dan tempat yaitu tidak dapat menyimpan atau mendokumentasikannya dalam data multimedia (foto, video, audio, dan note).

Pada Tugas Akhir ini, akan dikembangkan aplikasi *tracking* perjalanan berbasis Android yang berguna untuk melakukan perekaman dan mendokumentasikan perjalanan secara fleksibel dalam bentuk *multimedia* (foto, audio/voice, video dan note), sehingga seseorang dapat menyimpan *history* perjalanannya. *History* perjalanan juga dapat dilihat dan dibagikan kepada orang lain. Rekaman perjalanan yang dimaksud adalah hasil rekaman menggunakan sensor pada *smartphone*. Sensor yang digunakan seperti sensor audio untuk merekam suara pengguna, camera untuk merekam video dan mengambil foto dan GPS untuk melakukan *track* untuk menunjukkan letak titik koordinat tempat yang telah dilalui dan lokasi *handset*-nya. Tujuan pemanfaatan sensor dalam *smartphone* pada aplikasi ini adalah untuk mengumpulkan informasi tentang berbagai tempat yang dilalui dengan akurat dan tepat.

Dengan demikian pengembangan aplikasi dapat digunakan sebagai sarana untuk berbagi dan bertukar informasi perjalanan dengan orang lain, informasi perjalanan yang di maksud antara lain perjalanan ke tempat wisata, informasi perjalanan *survey* tempat atau lokasi untuk kepentingan tertentu, perjalanan ke tempat yang belum pernah dilalui ataupun untuk melakukan perekaman perjalanan sesuai dengan kemauan pengguna aplikasi. Apabila tempat wisata belum diketahui banyak orang maka informasi perjalanan ke lokasi wisata berupa *history* perjalanan yang telah disimpan dapat dibagikan kepada orang yang belum mengetahuinya, perjalanan *survey* lokasi atau tempat untuk kepentingan tertentu seperti untuk *survey* lokasi pembangunan jalan, gedung, lahan perkebunan atau pertanian dan lain-lain, seseorang dapat mengumpulkan informasi selama perjalanan maupun menuju tempat yang direncanakan dengan akurat serta dapat menunjukan gambaran perjalanan yang dilalui, selain itu aplikasi juga dapat digunakan untuk merekam dan menyimpan data perjalanan ke suatu tempat yang baru dikunjungi, sehingga data perjalanan tersebut dapat disimpan sebagai arsip perjalanan pengguna.

*Traditional development* memiliki sejumlah kelemahan yang membuatnya tidak cocok untuk proses dan situasi tertentu. Kelemahan paling serius adalah risiko tinggi dari proses dan sistem yang didukungnya sudah ketinggalan zaman jauh

sebelum diterapkan dalam praktik [6]. Hal ini terjadi karena *traditional development* membutuhkan waktu yang lebih banyak untuk melakukan proses pengembangan aplikasi secara *repetitive* karena *feedback* pengguna yang kurang tepat waktu. Permasalahannya adalah jika ada sebuah ide atau fungsi, bagaimana cara untuk menyelesaikannya secepat mungkin, bagaimana mendapatkan *feedback* secara cepat dan bagaimana mendapatkan *bug* secara cepat. Dalam siklus pengembangan perangkat lunak tradisional, integrasi perangkat lunak dilakukan jika suatu fase pengembangan perangkat lunak selesai. Hal ini akan menyebabkan integrasi perangkat lunak harus menunggu waktu yang lama dan dapat menyebabkan masalah saat integrasi, sehingga harus dilakukan perubahan terhadap aplikasi yang sudah selesai [7]. Untuk menyelesaikan permasalahan tersebut dibutuhkan suatu pendekatan yang dapat mengatasi masalah saat integrasi untuk mendeteksi kesalahan sedini mungkin sehingga tidak akan mengulang atau melakukan perubahan pada aplikasi yang sudah selesai. Pada pengembangan aplikasi ini akan menerapkan konsep *automation* pada saat implementasi yaitu *CICD (Continuous Integration Continuous Delivery)*. *CICD* adalah konsep *automation* yang akan memberikan kemudahan merelasikan kode program atau integrasi kode program, proses pembuatan dan pemantauan yang sederhana serta melakukan *deploy* di berbagai lingkungan yang diperlukan menggunakan *automation tools* untuk meningkatkan efisiensi pada pengembangan perangkat lunak [8]. Keuntungan yang dapat dihasilkan dari penerapan konsep *CICD* diantaranya mengurangi risiko, mengurangi proses manual yang berulang-ulang, menghasilkan perangkat lunak yang dapat dikembangkan dalam beberapa waktu dan beberapa tempat, memungkinkan visibilitas proyek yang lebih baik. Seperti pada perusahaan *marketplace* Tokopedia, mereka dituntut untuk selalu meningkatkan fitur website sehingga perusahaan tersebut menerapkan konsep *CICD*.

*Automation tools* yang akan dipakai pada pengembangan aplikasi ini adalah Jenkins. Jenkins adalah *platform* otomatisasi *open-source* yang banyak digunakan untuk membangun dan menggunakan perangkat lunak [9]. Ini adalah *plugin* yang sangat *extensible*, menyediakan *GUI (Graphical User Interface)* yang nyaman untuk membuat dan mengedit proses pembuatan. Jenkins akan membantu dalam mengembangkan dan menjalankan beberapa fitur-fitur aplikasi yang telah

ditentukan dan memeriksa pengembangan di perubahan perangkat lunak sehingga dapat menghemat waktu kerja pengembang dan meningkatkan efektivitas dan efisiensi pengembangan perangkat lunak.

## 1.2 Rumusan Masalah

Rumusan masalah yang menjadi acuan untuk pengerjaan dari Tugas Akhir ini adalah bagaimana mengembangkan aplikasi *tracking* perekaman perjalanan yang dapat merelasikan perjalanan (*journey*) dengan artefak multimedia (foto, video, voice record dan note) yang diperoleh sepanjang *journey*?

## 1.3 Tujuan

Tujuan dari Tugas Akhir ini adalah menghasilkan aplikasi *tracking* perjalanan berbasis *mobile* yang dapat merelasikan perjalanan (*journey*) dengan artefak multimedia (foto, video, voice record dan note) sehingga membentuk sebuah dokumentasi perjalanan dengan konsep CICD (*Continuous Integration Continuous Delivery*).

## 1.4 Lingkup

Lingkup pembahasan dalam kajian ini bertujuan agar masalah yang akan diteliti dapat lebih terarah dan terfokus sesuai dengan apa yang direncanakan yaitu:

1. Pengerjaan Tugas Akhir ini bertujuan untuk mengembangkan aplikasi yang dapat dimanfaatkan untuk *share* lokasi yang telah dilalui.
2. Dalam pengerjaan Tugas Akhir ini memanfaatkan API *google maps* sebagai alat untuk melacak lokasi dan menggunakan sensor pada *smartphone* untuk memperoleh data multimedia.
3. Pengerjaan Tugas Akhir ini dalam pengerjaannya akan menggunakan *platform* android dengan bahasa pemrograman Java.

## 1.5 Pendekatan

Metodologi penelitian dalam pengerjaan Tugas Akhir ini dilakukan menggunakan pendekatan Waterfall. Masalah yang ditemukan (spesifikasi kebutuhan) telah ditetapkan di awal dan sudah jelas. Fase-fase dalam pengembangan perangkat lunak

ini berurutan dan sistematis. Mulai dari *requirement* (analisis kebutuhan), *design system* (desain sistem), implementasi, penerapan/pengujian program dan pemeliharaan. Pada saat integrasi perangkat lunak dalam membutuhkan yang lama dan dapat menyebabkan masalah saat integrasi. Untuk mengatasi permasalahan tersebut, pada Tugas Akhir ini akan menerapkan konsep *automation* yaitu CICD yang akan digunakan pada saat proses implementasi perangkat lunak. Adapun tahapan dari pendekatan Waterfall pada Tugas Akhir ini adalah:

1. *Requirement* (analisis kebutuhan)

Tahap ini adalah tahap komunikasi yang bertujuan untuk pengumpulan dan pendefinisian ruang lingkup pengembangan masalah serta penentuan strategi yang akan digunakan pada topik Tugas Akhir. Pengumpulan informasi dalam Tugas Akhir ini dilakukan dengan diskusi. Pada tahap ini akan dicari informasi yang sebanyak-banyaknya dari *user* sehingga kebutuhan telah jelas didefinisikan di awal. Informasi tersebut dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna. Keluaran dari fase ini adalah dokumen *user requirement*. Dokumen inilah yang menjadi acuan untuk menerjemahkan menjadi ke dalam Bahasa pemrograman.

2. *Design System* (desain sistem)

Tahapan desain dilakukan untuk menerjemahkan syarat kebutuhan ke sebuah perancangan perangkat lunak. Tahapan desain yang dilakukan adalah desain basis data, pengajuan rancangan fungsi dan fitur yang akan diterapkan, arsitektur perangkat lunak dan gambaran desain dari antarmuka aplikasi yang akan dikembangkan. Keluaran dari tahap desain adalah *database*, *design interface*, arsitektur perangkat lunak dan kebutuhan *functional*.

3. Implementasi

Tahapan *coding* dilakukan untuk menerjemahkan kebutuhan yang diminta oleh *user*. Setiap fungsi akan diimplementasikan tim pengembang dan akan dilakukan *testing* terhadap sistem yang telah dibuat. Pada tahapan ini akan

diterapkan CICD pada saat integrasi kode program dari tim pengembang secara otomatis. Tahapan-tahapan dalam CICD adalah sebagai berikut:

*a. Integrated*

Merupakan tahapan dimana tiap-tiap developer mengintegrasikan kode-kode atau hasil kerja mereka.

*b. Compiled*

Kode dikompilasi kedalam paket atau executable

*c. Tested*

Melakukan uji coba executable secara manual atau otomatis

*d. Archived*

Mengarsipkan file executable, hasil tes, dan log selama proses tersebut berlangsung.

*e. Deployed*

Memasang atau membagikan hasil *build*.

4. Penerapan / Pengujian Program

Tahapan ini dilakukan setelah melakukan analisa, desain dan pengkodean maka sistem yang sudah jadi digunakan oleh *user*.

5. Pemeliharaan (*Operation & Maintenance*)

Tahap ini adalah tahap terakhir dalam pengembangan Tugas Akhir ini. Setelah sistem sudah jadi, akan dilakukan pemeliharaan untuk memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya.

## 1.6 Sistematika Penyajian

Sistematika penyajian Tugas Akhir ini menjadi tujuh bab, antara lain :

**Bab 1 Pendahuluan** menjelaskan mengenai gambaran tentang isi dari materi tugas akhir yang akan dikerjakan mulai dari latar belakang pengerjaan tugas akhir, ruang



lingkup dan tujuan dari topik akhir yang pilih, serta pendekatan yang dilakukan dalam menyelesaikan tugas akhir.

**Bab 2 Tinjauan Pustaka** menjelaskan mengenai dasar-dasar teori serta tinjauan pustaka yang berhubungan dengan topik tugas akhir yang dikerjakan.

**Bab 3 Analisis dan Desain** yang menggambarkan secara umum design dari aplikasi yang akan dikembangkan meliputi design database, UI/UX atau desain *interface*, desain fungsionalitas atau use case diagram, use case scenario dan yang lainnya.

**Bab 4 Implementasi** menjelaskan uraian hasil implementasi yang dilakukan.

**Bab 5 Pengujian** menjelaskan proses pengujian beserta rancangan yang telah dibahas pada bab sebelumnya

**Bab 6 Hasil dan Pembahasan** menjelaskan bagaimana fungsional aplikasi yang dikembangkan berjalan sesuai antara design dengan implementasi yang dikerjakan dan hasil dari evaluasi.

**Bab 7 Kesimpulan dan Saran** berisi mengenai kesimpulan yang diperoleh selama proses pengerjaan Tugas Akhir serta saran untuk perbaikan pengembangan aplikasi berikutnya.

## **BAB 2**

### **TINJAUAN PUSTAKA**

Pada bab tinjauan pustaka diuraikan dasar teori yang relevan dengan objek kajian Tugas Akhir. Tugas Akhir ini berkaitan dengan pengembangan aplikasi, maka pada bab ini akan dijelaskan domain permasalahan (*tracking*), GPS sebagai alat bantu untuk memperoleh data, dan sensor pada *smartphone* yang akan digunakan. Karena pengembangan aplikasi diimplementasikan pada *platform* berbasis android, maka akan dijelaskan dasar teori mengenai android dan basis data yang digunakan. Pada bab ini juga akan dibahas mengenai konsep CICD dengan *automation tools* yang akan digunakan yaitu Jenkins.

#### **2.1 Tracking**

*Tracking* menurut kamus Inggris – Indonesia memiliki arti mengikuti jalan, atau dalam arti bebasnya adalah suatu kegiatan untuk mengikuti jejak suatu objek. Salah satu cara yang dapat digunakan untuk melakukan pemantauan adalah dengan menggunakan GPS module. Keberadaan tempat berdasarkan posisi *latitude* dan *longitude* dapat diketahui dengan menggunakan GPS module, sehingga dari posisi *latitude* dan *longitude* tersebut dapat divisualisasikan dalam bentuk peta.

Tingkat kesulitan *tracking* objek tergantung pada pergerakan objek, perubahan pola objek dan latar belakang, mengubah struktur objek, halangan objek demi objek atau objek berdasarkan latar belakang, dan pergerakan kamera.

#### **2.2 GPS**

GPS adalah singkatan dari *Global Positioning System* yaitu sistem untuk menentukan navigasi dan posisi secara global dengan menggunakan metode Triangulasi dan satelit. Jumlah satelit GPS yang mengorbit bumi seluruhnya 24 buah, 21 buah aktif bekerja dan 3 buah sisanya adalah cadangan.

##### **1. Satelit (*Space Segment*)**

Satelit berfungsi untuk menyimpan data dan menerima daya yang ditransmisikan oleh stasiun-stasiun pengendali, menyimpan dan menjaga

informasi waktu dan memancarkan sinyal serta informasi secara berkelanjutan ke perangkat penerima (*receiver*).

## 2. Segmen pengendali (*Control Segment*)

Segmen pengendali berfungsi untuk mengendalikan satelit dari bumi mulai dari melihat keadaan satelit, penentuan serta prediksi orbit, sinkronisasi waktu antar satelit, dan mengirimkan data ke satelit.

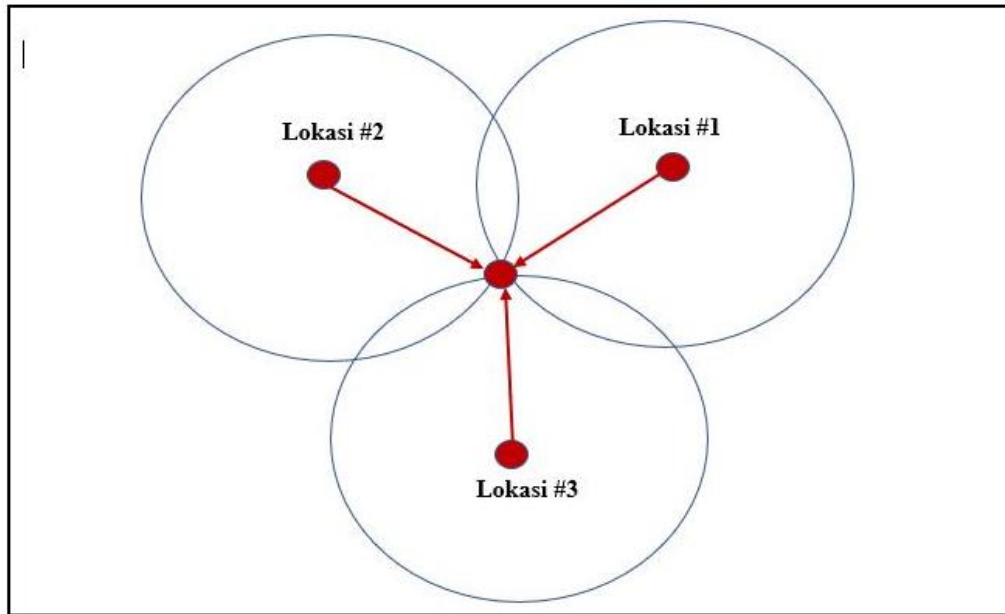
## 3. Segmen penerima (*User Segment*)

Segmen penerima berfungsi sebagai penerima data dari satelit dan memprosesnya untuk menentukan posisi, arah, jarak dan waktu yang diperlukan oleh pengguna.

Terdapat 5 langkah cara kerja GPS, yaitu [10]:

1. Memakai perhitungan “*triangulation*” dari satelit.
2. Perhitungan “*triangulation*”, yaitu GPS mengukur jarak menggunakan travel time sinyal radio.
3. GPS memerlukan akurasi waktu yang tinggi yang digunakan untuk mengukur *travel time*.
4. Posisi satelit dan ketinggian pada orbitnya harus diketahui dengan pasti yang digunakan untuk perhitungan jarak,
5. Langkah terakhir yaitu mengoreksi penundaan sinyal waktu perjalanan di atmosfer sampai diterima *receiver*.

Konstelasi 27 buah satelit yang mengorbit bumi digunakan GPS, dimana untuk menentukan posisi, sebuah GPS *receiver* harus menerima informasi dari tiga atau lebih satelit [10]. GPS *receiver* akan berguna menerima sinyal dari GPS dan posisi diubah menjadi titik (*Way-Point*) yaitu titik-titik koordinat bujur dan lintang dari posisi atau lokasi. Penentuan titik dapat dilihat pada Gambar 2.1.



**Gambar 2.1. Trilaterasi Dalam Global Positioning System (GPS)**

*(Sumber : Deteksi Gempa Dengan Piezo Elektrik Berbasis Mikrokontroler, 2013)*

Layanan penggunaan dari GPS antara lain :

- a. Lokasi dapat digunakan untuk menentukan titik atau lokasi keberadaan.
- b. Navigasi dapat digunakan untuk mencari suatu lokasi suatu titik di bumi.
- c. *Tracking* dapat digunakan untuk memantau pergerakan suatu objek dan menentukan posisi tertentu dan perhitungan jaringan terdekat.
- d. *Timing* dapat digunakan untuk penentuan jam dan kebutuhan untuk aplikasi yang membutuhkan presisi waktu dalam suatu sistem.
- e. *Mapping* dapat digunakan untuk digunakan untuk memetakan suatu tempat atau lokasi.

### 2.3 Sensor

Sensor adalah dari perangkat keras (*hardware*) yang ditanamkan dalam perangkat yang berfungsi untuk mengirim data dari lingkungan fisik ke aplikasi. Sensor pada android juga dinyatakan sebagai kemampuan yang dapat merekam pengukuran terhadap perangkat dan lingkungan eksternalnya.

Jenis sensor pada perangkat android antara lain [11]:

- a. Sensor Lokasi yaitu digunakan untuk menentukan lokasi atau posisi perangkat menggunakan berbagai sensor termasuk GPS.
- b. Sensor Fisik yaitu digunakan untuk mendeteksi sifat perangkat secara spesifik seperti orientasi, akselerasi, dan rotasi beserta sifat lingkungan seperti cahaya, medan magnet, dan tekanan udara.
- c. NFC Scanner yaitu digunakan untuk mendeteksi *tag Near Field Communication* (NFC) terdekat dan berbagi data dengan NFC perangkat android lainnya yang aktif.
- d. Kamera yaitu digunakan untuk mengumpulkan gambar visual.
- e. Mikrofon yaitu digunakan untuk merekam suara.
- f. *Speech recognition* yaitu digunakan untuk mengubah suara menjadi teks dengan memakai gabungan dari algoritma *recognition* dan suara hasil rekaman mikrofon.
- g. Sensor eksternal yaitu setiap sensor yang terhubung melalui mekanisme *Android Open Accessory* (AOA).

Sensor yang dipakai pada pengembangan aplikasi ini adalah sensor lokasi untuk menentukan lokasi yang akan dijalani, sensor kamera untuk merekam *history* perjalanan, dan mikrofon digunakan merekam suara pengguna untuk memulai dan memberhentikan aplikasi.

## 2.4 Android

Aplikasi yang akan dikembangkan ini pada perangkat *mobile* menggunakan sistem operasi Android. Platform Android dipilih karena merupakan sistem operasi yang paling banyak digunakan oleh pengguna *smartphone*. Tujuannya agar pengguna dapat merekam *history* perjalanan menggunakan *smartphone* yang mudah dibawa kemana-mana. Android adalah modifikasi dari sistem operasi Linux menjadi sistem operasi bergerak (*mobile operating system*). Pada tahun 2005, Google mengambil alih Android dari Android, Inc untuk mengisi pasar sistem operasi bergerak sebagai

bagian strategi. Seluruh hasil kerja Android termasuk tim yang mengembangkan Android diambil alih oleh Google. Google menginginkan agar Android bersifat terbuka dan gratis yang berarti bahwa semua orang yang ingin menggunakan Android dapat men-*download* penuh source *code*-nya., oleh karena itu hampir setiap kode program Android diluncurkan berdasarkan lisensi *open-source* Apache. Adanya pendekatan aplikasi secara terpadu adalah keuntungan utama dari Android. Aplikasi bisa dijalankan pada beberapa perangkat yang berbeda selama masih diberikan tenaga oleh Android (pengembang tidak perlu mempertimbangkan kebutuhan jenis perangkatnya).

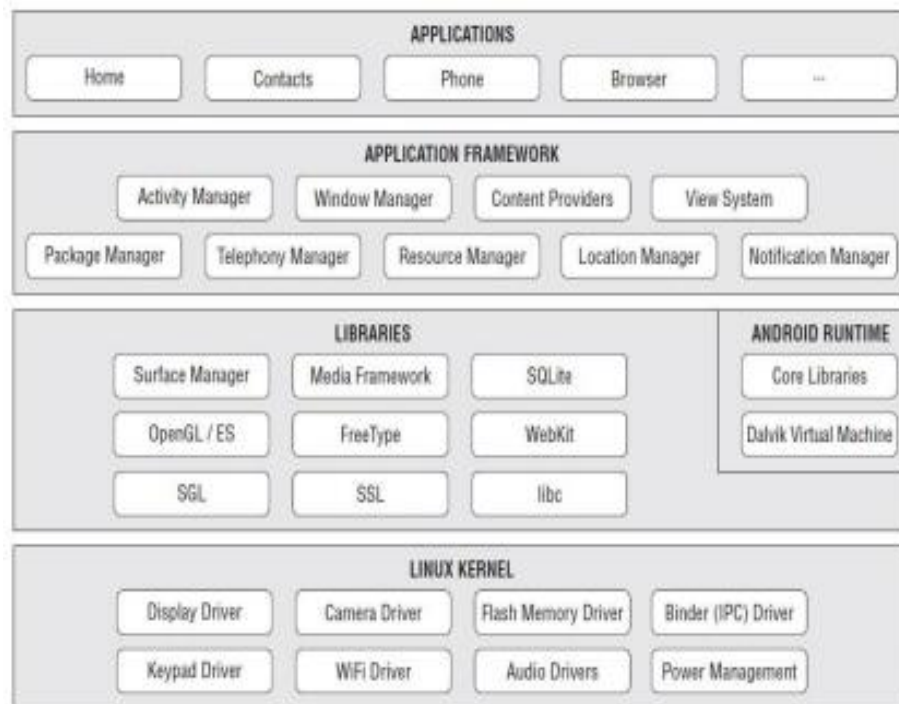
Versi – versi Android dapat dilihat pada Tabel 2.1.

**Tabel 2.1. Versi Android**

(Sumber : telah diolah kembali dari : [https://id.wikipedia.org/wiki/Daftar\\_versi\\_Android](https://id.wikipedia.org/wiki/Daftar_versi_Android))

Versi Android	Nama Kode	Tahun
Android 1.0	Apple Pie	2008
Android 1.1	Banana Bread	2009
Android 1.5	Cupcake	2009
Android 1.6	Donut	2009
Android 2.0	Eclair(API 5)	2009
Android 2.1	Eclair(API 6)	2009
Android 2.01	Eclair(API 7)	2010
Android 2.29	Froyo	2010
Android 2.3	GingerBread	2010
Android 3.0-3.26	Honeycomb	2011
Android 4.0	Ice Cream Sandwich	2011
Android 4.12	Jelly Bean	2012
Android 4.4	Kitkat	2013
Android 5.0	Lolipop	2014
Android 6.0	Marshmallow	2015
Android 7.0	Nougat	2016
Android 8.0	Oreo	2017
Android 9.0	Pie	2018

Berikut ini Gambar 2.2 bagan tingkatan-tingkatan sistem operasi Android [12].



**Gambar 2.2. Arsitektur Sistem Operasi Android**

(Sumber : Pemrograman Aplikasi Android , 2013)

Pada umumnya sistem operasi Android terbagi menjadi lima tingkatan yaitu :

1. Linux kernel adalah kernel dasar Android. Tingkat ini berisi semua driver perangkat tingkat rendah untuk komponen-komponen hardware perangkat Android.
2. *Libraries* berisi semua kode program yang menyediakan layanan-layanan utama sistem operasi Android.
3. Android Runtime berguna untuk menyediakan kumpulan pustaka inti yang dapat diaktifkan oleh pengembang untuk menulis kode aplikasi Android dengan bahasa pemrograman Java.
4. *Application framework* adalah kumpulan class *built-in* yang tertanam dalam sistem operasi Android yang dapat dimanfaatkan oleh pengembang untuk aplikasi yang sedang dikembangkan.
5. *Applications* untuk tempat pengguna akan bekerja, contoh aplikasi ini banyak ditemui, seperti: *Phone*, *Contact*, *Browser* dan lain-lain.

Seperti aplikasi Android pada umumnya yang dapat *download* dan *install* dari Market Android. Semua aplikasi yang dibuat pengguna terletak pada tingkat Aplikasi.

Pada Tugas Akhir ini akan digunakan Android Studio sebagai IDE (*Integrated Development Environment*). Android Studio dirilis 16 May 2013 pada saat Google I/O 2013. Android Studio berbasis JetBrains IntelliJ IDEA, dan dikhususkan untuk mengembangkan *software Platform* Android [13].

#### 2.4.1 Java Programming Language

Bahasa pemrograman yang dapat digunakan untuk *development* Android antara lain Java, C/C++, Go, Kotlin, dll [13]. Java menyediakan fitur untuk mendefinisikan objek dan mengakses properti pada *code* serta fitur untuk mengirim pesan. Contohnya melakukan perubahan pada objek untuk melakukan beberapa hal seperti menggunakan objek kamera untuk mengambil gambar. Bahasa pemrograman Java cocok dengan banyak perangkat (termasuk perangkat *mobile*) dan mengabstraksi kerumitan pada sistemnya. Android merupakan aplikasi berbasis Java yang menggunakan Dalvik *Virtual Machine* (DVC) untuk menerjemahkan dan mengeksekusi *code* Java dan dioptimasi untuk dioperasikan pada *Platform mobile* [14]. Aplikasi akan dikembangkan dengan bahasa pemrograman Java.

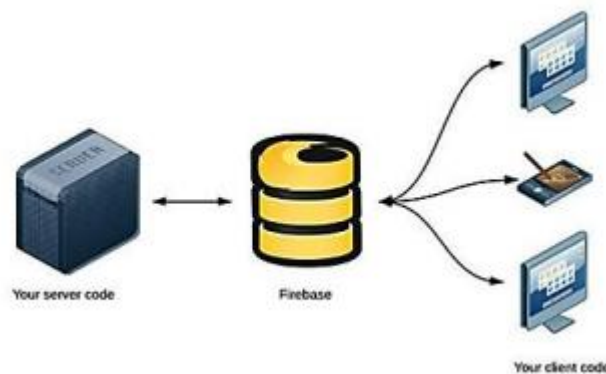
#### 2.4.2 Firebase

Basis data yang akan digunakan untuk membangun aplikasi pada *platform* Android adalah *firebase*. Hal ini dikarenakan *firebase* merupakan solusi untuk mempermudah dalam proses pengembangan aplikasi *mobile* yang ditawarkan oleh Google. Firebase database merupakan penyimpanan basis data non-SQL yang memungkinkan untuk menyimpan beberapa tipe data antara lain String, Boolean, Long, dan Boolean. *Firebase* juga dapat menyimpan file seperti file foto, file video, *textfile* dan MP3. Data pada Firebase database disimpan sebagai objek JSON tree. Yaitu pada basis data non-SQL tidak ada tabel dan baris, tidak seperti basis data SQL. Ketika ada penambahan data, data tersebut akan menjadi node pada struktur struktur JSON [15]. Database yang menyediakan API untuk disinkronisasikan kepada *client* yang berbeda dan disimpan pada cloud-nya *firebase* merupakan produk utama dari *firebase*. Banyak *library* yang dimiliki oleh



*Firebase* yang memungkinkan untuk merelasikan layanan ini dengan Android, Ios, Javascript, Java, Object-C dan Node.JS. *firebase client library* yang sudah diterapkan pada aplikasi yang dikembangkan yang akan mengambil data secara *realtime* [16].

Pada Gambar 2.3 menunjukkan arsitektur dari sistem *firebase*.



**Gambar 2.3. Arsitektur Sistem Firebase**

(Sumber : Aplikasi E-Order Menggunakan Firebase dan Algoritma Knuth Morris Pratt Berbasis Android, 2018)

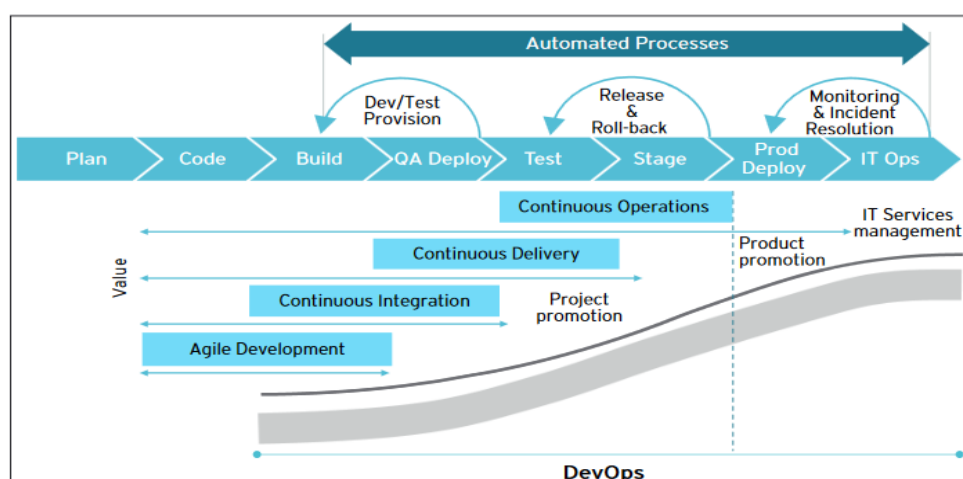
## 2.5 Google Maps API

*Google Maps API* merupakan layanan (*service*) yang diberikan oleh Google untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi kepada para pengguna. Beberapa fitur untuk memanipulasi peta, menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi di dalam websitenya disediakan *Google Maps API*. Layanan-layanan yang ditawarkan oleh *Google Maps* dapat dimanfaatkan oleh pengguna setelah melakukan registrasi dan mendapatkan *Google Maps API Key*. [17]

## 2.6 CICD (Continuous Integration Continuous Delivery)

*Continuous Integration* (CI) adalah proses pengujian dan pembuatan *software* secara otomatis setelah kode aplikasi yang baru merelasikan ke dalam *repository* (tempat penyimpanan) bersama. Sedangkan *Continuous Delivery* (CD) adalah proses penyampaian aplikasi yang dibuat dalam proses CI ke bagian lingkungan produksi, yang dimasukkan melalui *automated test*. *Automated test* memastikan fungsi aplikasi sesuai dengan harapan dan kebutuhan ketika

didorong ke lingkungan produksi hingga ke tangan pengguna yang sebenarnya (*real users*) [18]. CI terjadi pada saat tim melakukan kode untuk merelasikan setiap hari. Selama merelasikan, perangkat lunak akan melalui pengujian dan pembuatan otomatis. Proses dasar CI adalah pengembang menyelesaikan tugas di komputernya dan melakukan perubahan di *repository* kontrol, sistem CI mengambil perubahan terakhir dalam *repository* secepat mungkin, sistem CI melakukan pembuatan perangkat lunak berdasarkan pada skrip pembuatan, setelah pembangunan, sistem CI memberikan informasi perubahan dan hasil pembangunan. *Continuous Delivery* adalah pengembangan perangkat lunak yang memungkinkan perangkat lunak siap untuk ditempatkan dalam produksi kapan pun dan dimana pun. Tujuan utama CICD (*Continuous Integration Continuous Delivery*) adalah mengurangi resiko, proses manual yang berulang-ulang, menghasilkan perangkat lunak yang dapat digunakan kapan saja dan dimana saja, memungkinkan visibilitas proyek yang lebih baik, membangun kepercayaan yang lebih besar pada produk perangkat lunak dari tim *developer* [19]. Pada pengembangan aplikasi ini akan menerapkan konsep CICD yang akan memberikan kemudahan merelasikan *code*, proses pembuatan dan pemantauan yang sederhana serta melakukan *deploy* di berbagai lingkungan yang diperlukan menggunakan *automation tools* untuk meningkatkan efisiensi dan efektivitas pengembangan aplikasi pada saat implementasi. Pada Gambar 2.4 menunjukkan arsitektur dari CICD.



**Gambar 2.4. Arsitektur sebuah CICD**

(Sumber : *Continuous Integration Improving Software Quality and Reducing Risk*, 2007)

### 2.6.1 Jenkins

Jenkins merupakan salah satu tool CI yang dikembangkan oleh Kohsuke Kawaguchi dengan nama awal Hudson pada akhir 2004 ketika masih bekerja di Sun. Hudson berkembang selama beberapa tahun, sistem ini diadopsi oleh banyak tim dalam Sun untuk proyek-proyek mereka. Pada tahun 2010, Hudson telah menjadi *tools* paling diminati dalam solusi *Continuous Integration* dengan pangsa pasar lebih dari 70%. Pada bulan Januari 2011, komunitas pengembang Hudson memutuskan untuk mengubah nama proyek menjadi Jenkins [20]. Jenkins adalah *platform* otomatisasi *open-source* yang banyak digunakan untuk membangun dan menggunakan perangkat lunak karena menyediakan *plugin* yang sangat extensible, dan menyediakan GUI yang nyaman untuk membuat dan mengedit proses pembuatan. Berikut kegunaan dari Jenkins antara lain sebagai berikut :

1. Seringnya rilis perangkat lunak baru karena setiap komit yang dibuat dalam sumber dibuat dan diuji.
2. Pengembang mengetahui hasil tes setiap komit yang dibuat dalam kode sumber saat dijalankan.
3. Pengembang hanya perlu melakukan perubahan pada kode sumber dan Jenkins akan mengotomatiskan seluruh proses.

Pada Gambar 2.5 menunjukkan konfigurasi dan *job list* dari Jenkins.



**Gambar 2.5. Jenkins Job List and Configurations**

(Sumber : Automatic Deployment System Dengan Menggunakan Metode Continuous Integration  
Di Kakatu)

### 2.6.2 Instalasi Jenkins dan Konfigurasi *Repository* dengan Jenkins

Pada sub bab ini dijelaskan mengenai tahapan dilakukan untuk melakukan instalasi Jenkins dan konfigurasi *repository* dengan Jenkins. Langkah-langkah yang dilakukan antara lain :

a. Instalasi Jenkins

Langkah pertama yang dilakukan adalah melakukan instalasi Jenkins.

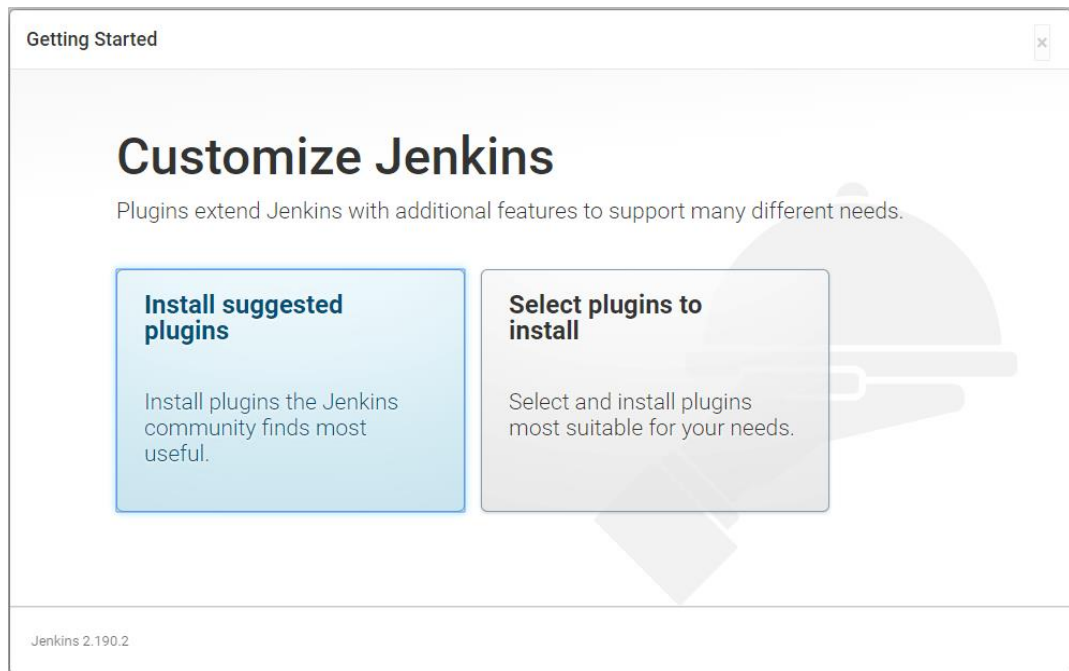
b. *Setting Up* Jenkins

Setelah instalasi selesai, halaman web yang berjalan pada localhost akan terbuka otomatis seperti pada Gambar 2.6. Pada halaman ini dibutuhkan *unlock key* untuk yang diletakkan pada *password* administrator untuk membuka Jenkins yang masih dikunci.



**Gambar 2.6. Webpage Running**

Langkah selanjutnya adalah melakukan instalasi plugin seperti git, subversion, gradle, dll pada halaman 'Customize Jenkins' seperti pada Gambar 2.7.



**Gambar 2.7. Customize Jenkins**

Setelah semua *plugin* di install selesai, maka langkah selanjutnya adalah membuat *user* admin pertama seperti pada Gambar 2.8.

**Gambar 2.8. Create First Admin User**

Setelah *user* berhasil dibuat akan masuk ke halaman Sign In untuk masuk ke Jenkins seperti pada Gambar 2.9.



**Welcome to Jenkins!**

Username

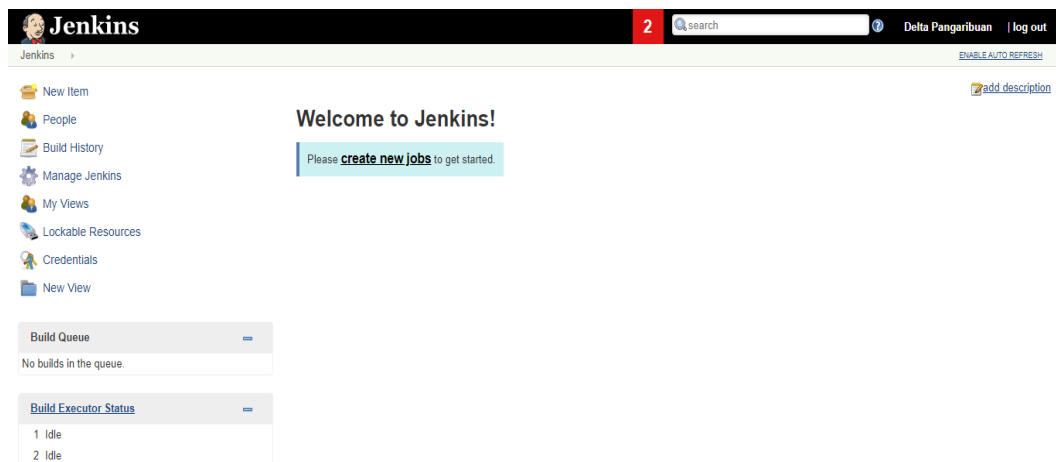
Password

**Sign in**

☐ Keep me signed in

**Gambar 2.9. Sign In**

Setelah berhasil masuk, akan muncul tampilan utama Jenkins seperti pada Gambar 2.10.



**Gambar 2.10. Jenkins Dashboard**

c. Konfigurasi Jenkins untuk Android

Untuk membangun proyek Android di Jenkins, beberapa hal yang diperlukan yaitu Git, JDK, Gradle, AndroidSDK.

Pada Gambar 2.11 ini akan melakukan instalasi JDK dan Setting up JDK dan Android\_SDK.

**Global properties**

☐ Disable deferred wipeout on this node ?

☒ Environment variables

List of variables

Name	JAVA_HOME	
Value	C:\Program Files\Java\Java JDK 10.0.2\bin\java.exe	Delete
Name	ANDROID_HOME	
Value	C:\Program Files\android_sdk	Delete

Add

☐ Tool Locations

**Pipeline Speed/Durability Settings**

Pipeline Default Speed/Durability Level: None: use pipeline default (MAX\_SURVIVABILITY) ?

**Usage Statistics**

☒ Help make Jenkins better by sending anonymous usage statistics and crash reports to the Jenkins project. ?

Save Apply

**Gambar 2.11. Jenkins Environment Variables**

d. Konfigurasi sebuah proyek Android di Jenkins

Pertama sekali yaitu membuat suatu *job* atau item baru seperti pada Gambar 2.12.

**Jenkins** 2 search Delta Pangaribuan

Jenkins > All >

**Enter an item name**

TA-D3T13

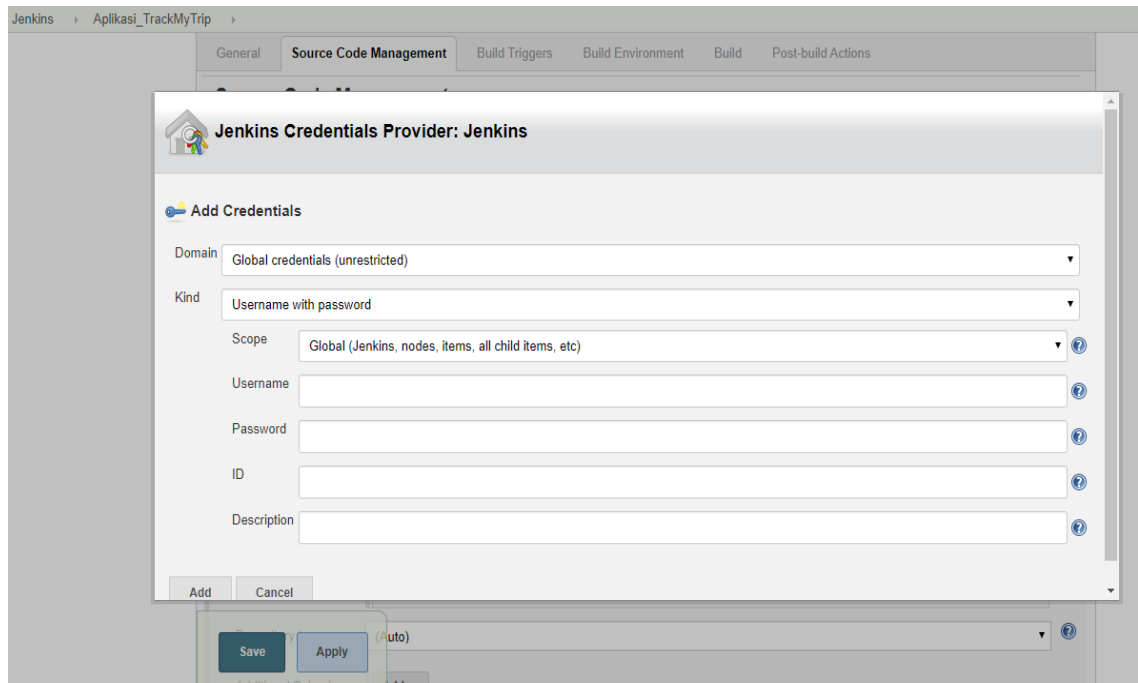
» A job already exists with the name 'TA-D3T13'

- Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**  
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multi-branch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

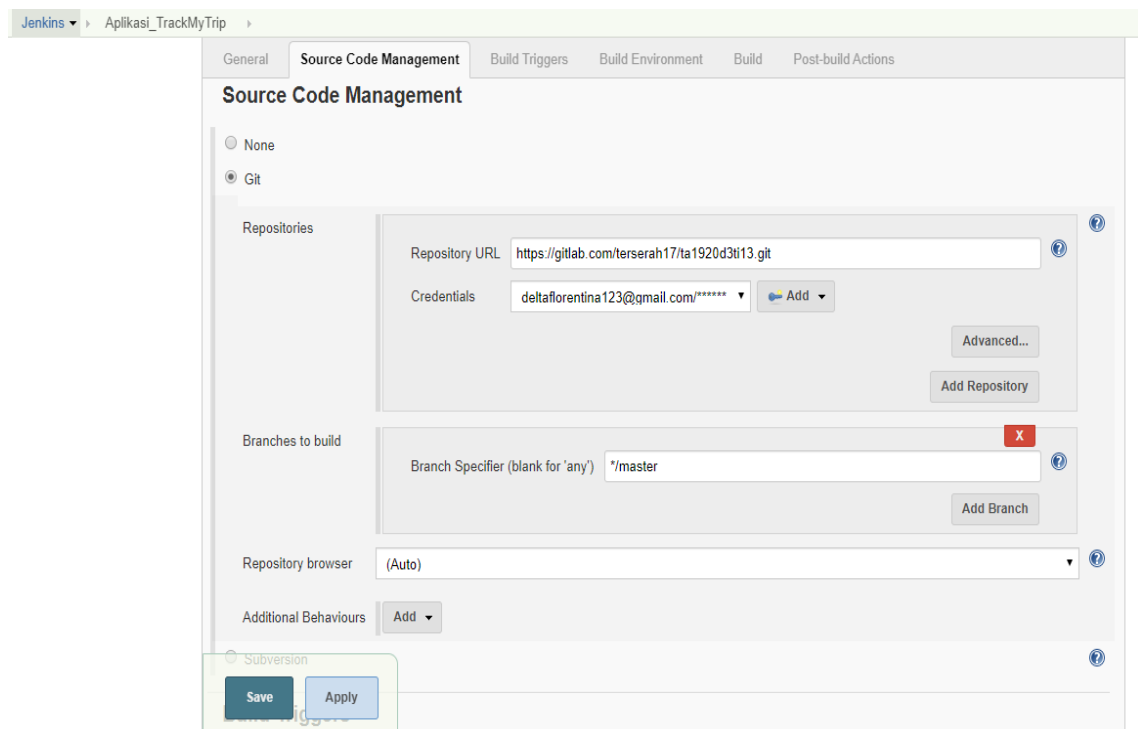
**Gambar 2.12. Jenkins Job Style**

Langkah selanjutnya adalah mengatur repositori Git yang akan dilakukan operasi *build* seperti pada Gambar 2.14. Pada tab Source Code Management, akan

dimasukkan credentials yaitu *username* dan *password* yang digunakan untuk Git seperti pada Gambar 2.13.



**Gambar 2.13. Jenkins Credentials Provider**

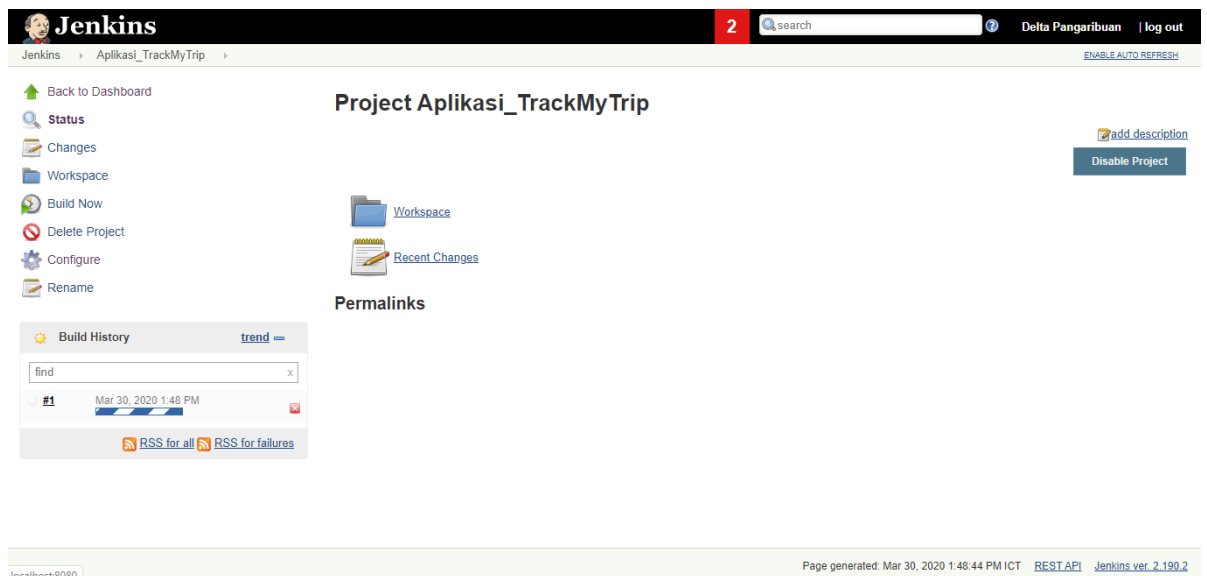


**Gambar 2.14. Jenkins Source Code Management**



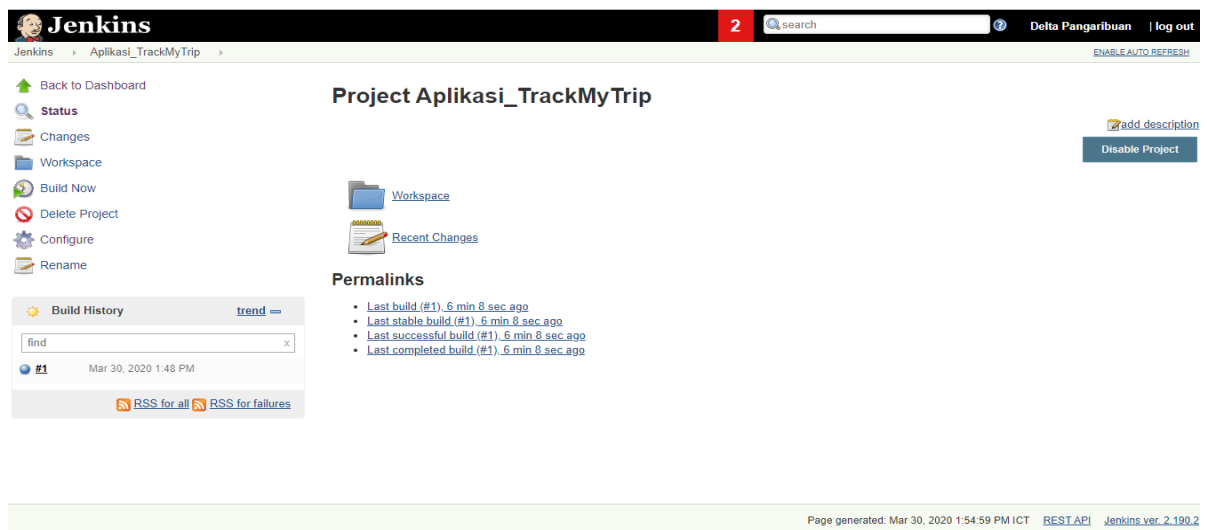
e. *Build* proyek Android

Langkah selanjutnya adalah melakukan *build* pada proyek android yang telah dibuat sebelumnya. Pada Dashboard, lakukan opsi Build Now. Maka akan muncul proses *running* di Build History seperti pada Gambar 2.15.



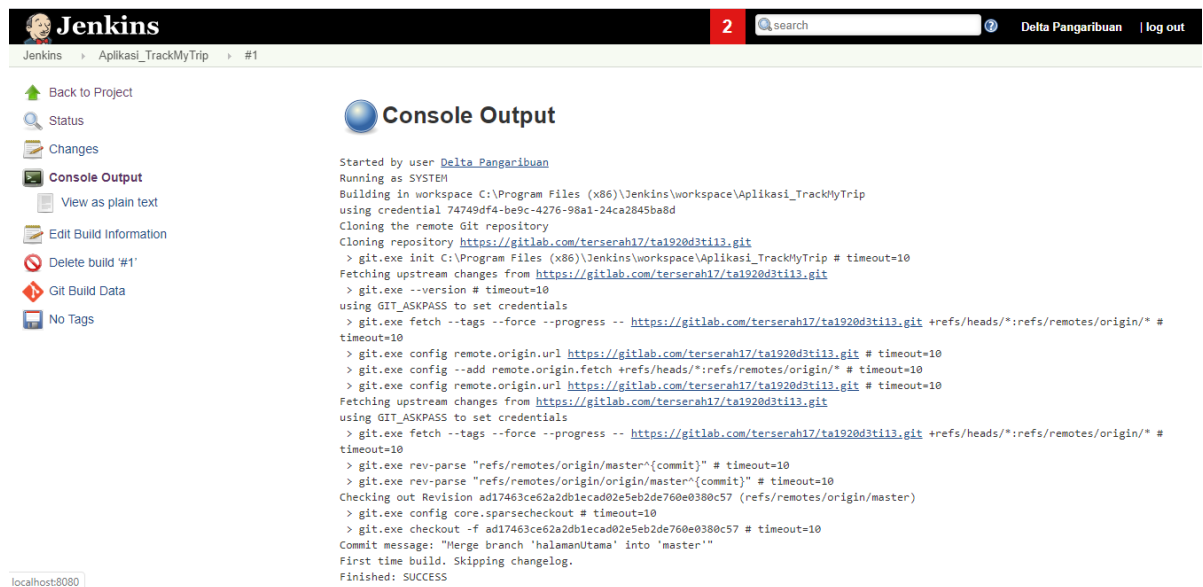
Gambar 2.15. Process Build

Setelah proses *build* selesai, *dot color* (warna titik) akan berubah. Jika terjadi kesalahan maka *dot color* (warna titik) akan berubah menjadi merah, tetapi jika proses *build* berhasil maka *dot color* (warna titik) berubah menjadi biru. Hasil *build* dapat dilihat pada Gambar 2.16.



Gambar 2.16. Success Build

Pada Gambar 2.17 dapat dilihat hasil *build* melalui *console output*.



Gambar 2.17. Jenkins Console Output

## 2.7 Kesimpulan

Pada saat ini *tracking* sangat dibutuhkan karena memiliki berbagai manfaat dalam kehidupan manusia. *Tracking* merupakan suatu kegiatan untuk mengikuti sebuah objek. *Tracking* membutuhkan suatu teknologi untuk menentukan posisi yaitu GPS. Seiring perkembangan teknologi saat ini, aplikasi *tracking* dengan berbagai fungsi dan kegunaan masing-masing telah banyak beredar. Pada Tugas Akhir ini akan mengembangkan aplikasi yang dapat merekam *history* perjalanan seseorang secara fleksibel dan menyimpannya dalam bentuk data multimedia. Aplikasi akan dikembangkan pada *platform* yaitu Android. Pengembangan aplikasi Android ini harus memperhatikan setiap aspek yang diperlukan dalam proses pembangunan aplikasi Android, aspek yang diperhatikan seperti pemanfaatan sensor pada *smartphone* pemanfaatan aplikasi pendukung lainnya ataupun aplikasi pendukung yang digunakan seperti GPS, *firebase*, *tracking*, *platform* yang digunakan, bahasa program yang akan diimplementasikan, pemanfaatan fungsi sensor dalam *smartphone* dan penerapan konsep CICD dengan benar dan sesuai agar dapat meningkatkan efisiensi pengembangan perangkat lunak.

## **BAB 3**

### **ANALISIS DAN DESAIN**

Pada bab ini diuraikan hasil analisis dan desain kebutuhan penelitian yang menjadi yang menjadi acuan pada tahap implementasi. Analisis bertujuan untuk mengetahui spesifikasi kebutuhan dan rancangan aplikasi yang dikembangkan berdasarkan analisis dari aplikasi sejenis, analisis aplikasi *tracking*, analisis teknologi, kebutuhan pengguna terhadap aplikasi dan target sistem.

#### **3.1 Analisis Navigasi Tracking**

*Travel* atau perjalanan merupakan suatu peristiwa perpindahan tempat dari suatu titik lokasi ke lokasi yang lainnya secara bertahap atau berskala. *Tracker* berarti lacak atau melakukan monitoring terhadap suatu benda, berupa monitoring lokasi dan perpindahan tempat. Aplikasi *tracking* merupakan serangkaian kegiatan yang dilakukan untuk monitoring perpindahan, merekam perjalanan yang dilakukan oleh benda atau manusia [21]. Aplikasi akan digunakan untuk semua perjalanan, baik itu perjalanan pariwisata, survei, naik gunung, dll oleh berbagai golongan masyarakat. Pada militer, *track* digunakan untuk keperluan perang, seperti menuntun arah bom, atau mengetahui posisi pasukan berada. Pada umumnya aplikasi *track* digunakan untuk mencari seseorang, suatu tempat, ataupun benda. Ada juga aplikasi *track* yang mencatat lintasan lari, waktu yang digunakan, jarak ketika kita berolahraga. Aplikasi *track* dibutuhkan oleh wisatawan agar mereka tahu lokasi-lokasi wisata yang akan di kunjungi. Aplikasi *track* dibutuhkan bagi orang-orang yang ingin melakukan *survey* di daerah terpencil untuk mencatat lintasan perjalanannya. Aplikasi tracker juga dibutuhkan oleh pemilik kendaraan, agar perjalanan dari kendaraan tersebut dapat dilacak dan agar dapat dilacak ketika kendaraan tersebut hilang.

Pada umumnya terdapat beberapa domain yang menjadi basis dalam pembangunan aplikasi *tracking* sendiri, Track My Trip melakukan perekaman rute perpindahan tempat, dimana dalam perpindahan tersebut terdapat jarak, titik lokasi berupa koordinat (*longitude, latitude*), waktu yang dilakukan dalam perpindahan selain itu dalam melakukan perjalanan seseorang juga dapat melihat perjalanan yang dilalui

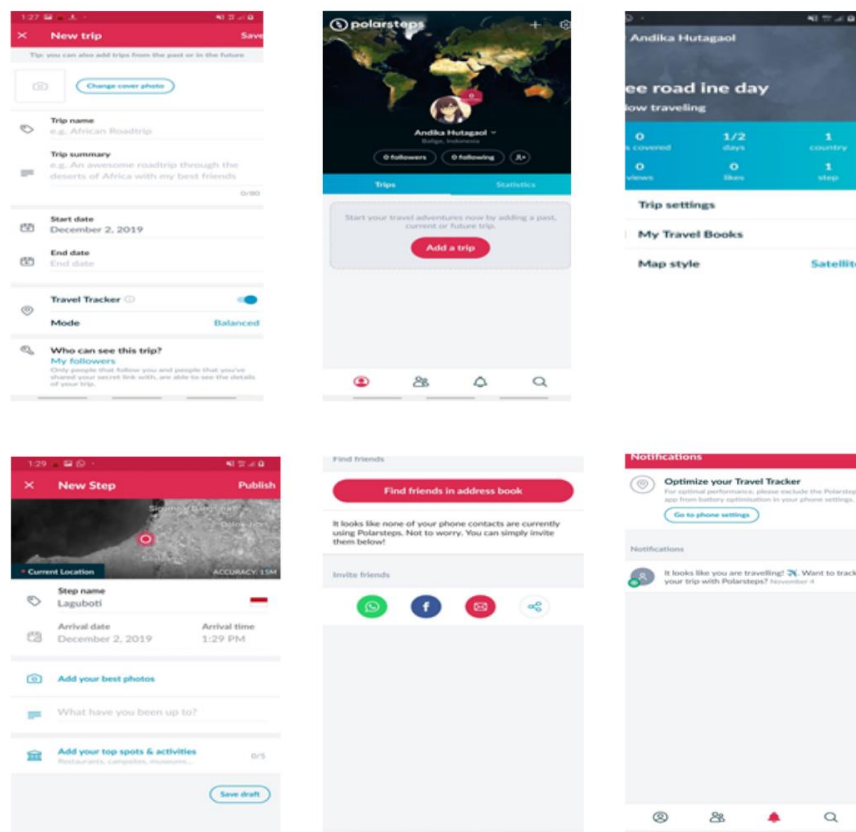
berupa tempat atau lokasi yang menarik, yang dapat didokumentasikan berupa foto, audio, video dan note saat berada pada suatu tempat tertentu, dengan demikian apabila seseorang mendokumentasikan suatu tempat maka data tersebut dapat disimpan sebagai data yang bermanfaat sebagai data dokumentasi perjalanan. Maka pengguna pun dapat melihat jurnal perjalanan yang indah nantinya.

### **3.2 Analisis Aplikasi *Tracking* Sejenis**

Pada sub bab ini dijelaskan mengenai hasil analisis terhadap aplikasi *tracking* sejenis yang menyediakan fitur umum pada aplikasi *tracking*. Aplikasi *tracking* tersebut adalah PolarSteps, Rute Saya, Google Maps, Maverick, GPX Viewer and Recorder, Jejak Kaki.

#### **3.2.1 PolarSteps**

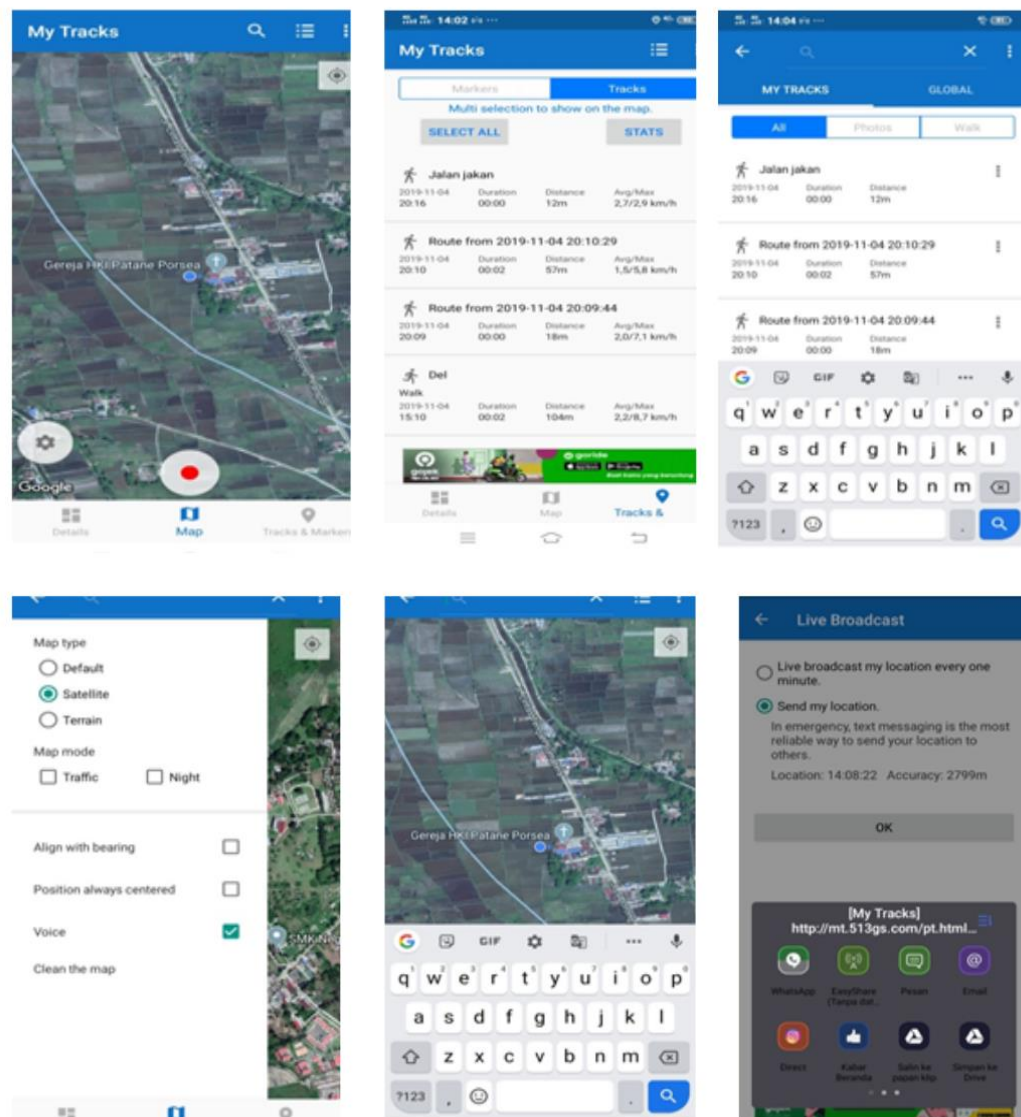
PolarSteps merupakan aplikasi *tracking* yang secara otomatis melacak rute dan tempat-tempat yang dikunjungi selama perjalanan. Setelah terhubung ke internet, rute perjalanan, lokasi, foto-foto akan ditransfer ke profil pelancong pribadi, dan rute pengguna dicantumkan pada peta dunia penuh warna yang bisa dibagikan dengan mudah kepada teman dan keluarga. Di akhir perjalanan, kita bisa langsung memesan sebuah album foto cetak nan indah. Pada aplikasi ini juga pengguna dapat membagikan pengalaman di luar pengguna aplikasi dan dapat mengirimkan secara langsung ke *Instagram*, *Facebook*, *Whatsapp*, dan media sosial lainnya. Untuk fungsi aplikasi sama seperti aplikasi *tracking* lainnya yaitu menunjukkan lokasi keberadaan pengguna dan menunjukkan peta. Pada Gambar 3.1 menunjukkan tampilan dari aplikasi PolarStep.



Gambar 3.1. Tampilan Polarstep

### 3.2.2 Rute Saya

Rute Saya adalah aplikasi kecil dan kuat untuk melacak rute pengguna saat pengguna berkeliling. Fungsionalitas yang cukup kompleks bersembunyi di balik antarmuka pengguna yang sangat jelas yang mudah dimengerti. Rute Saya dapat sangat berguna untuk semua kegiatan di luar ruangan seperti hiking, sepeda dan *tour* sepeda motor, berperahu, ski, panjat tebing, atau kesenangan berkendara semata, juga dapat digunakan untuk bisnis. Fitur yang ada antara lain rekam rute dapat menunjukkan lokasi saat ini di Google Map, bersama dengan waktu, durasi dan jarak dengan lintang dan bujur, rute merekam, menjeda, melanjutkan, menyimpan dan mendaftar. Foto bergabung dengan rute secara otomatis, laporan suara pada frekuensi waktu atau jarak yang telah ditentukan saat merekam. Share rute ke file GPX/KML/KMZ, atau impor dari ponsel pengguna atau Google Drive. Untuk fungsi aplikasi sama seperti aplikasi *tracking* lainnya yaitu menunjukkan lokasi keberadaan pengguna dan menunjukkan peta. Pada Gambar 3.2 menunjukkan tampilan dari aplikasi Rute Saya.

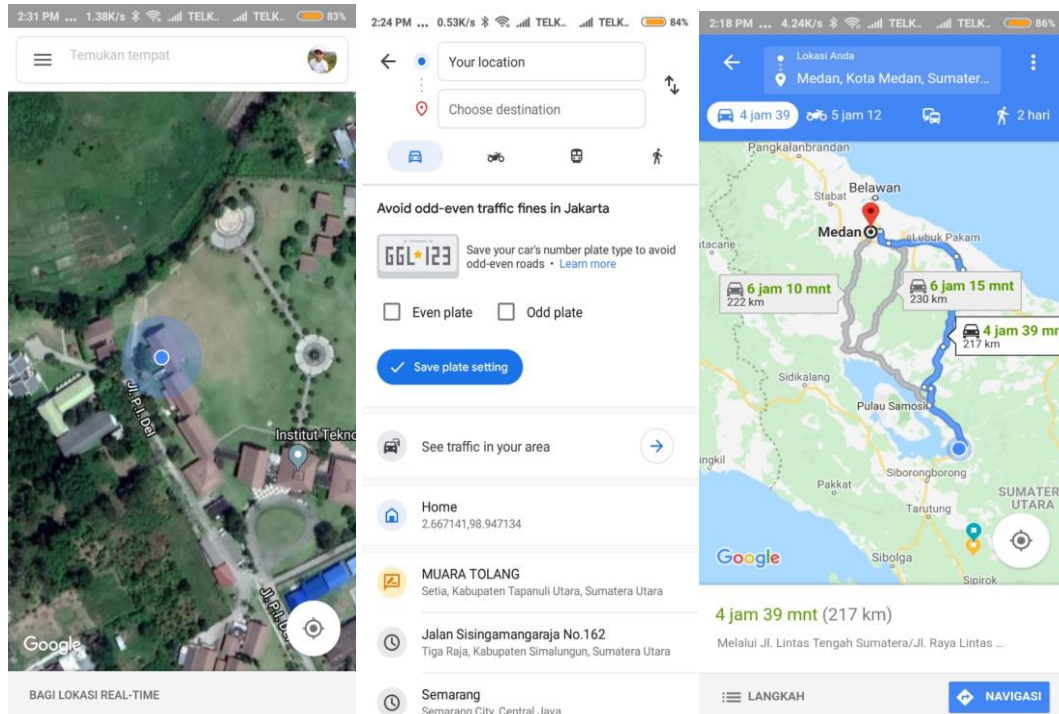


Gambar 3.2. Tampilan RuteSaya

### 3.2.3 Google Maps

Maps adalah aplikasi navigasi yang cepat dan mudah. Pengguna dapat menggunakan aplikasi untuk mempermudah pergi ke tempat tujuan dengan update *real-time*, menemukan aneka tempat dan jelajahi kota seperti acara, tempat makan dan aktivitas yang menarik, mencari tahu tempat baru dan sedang populer, dan dapat melihat daftar tempat favorit yang dapat dibagikan dengan teman dengan memakai Google Maps. Untuk fungsi aplikasi sama seperti aplikasi *tracking*

lainnya yaitu menunjukkan lokasi keberadaan pengguna dan menunjukkan peta. Pada Gambar 3.3 menunjukkan tampilan dari aplikasi Google Maps.

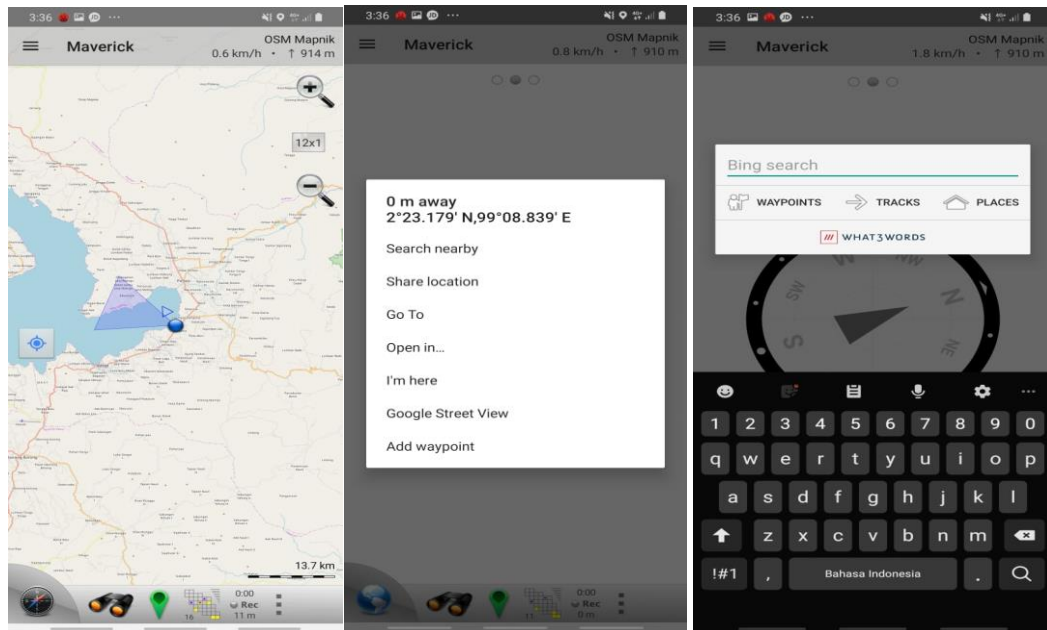


Gambar 3.3. Tampilan GoogleMaps

### 3.2.4 Maverick

Maverick merupakan aplikasi yang dapat digunakan untuk mendaki, berperahu, berpetualang dan kegiatan diluar lainnya. Dengan aplikasi ini, pengguna dapat membagikan posisi saat ini atau yang direncanakan, menyimpan tempat yang dikunjungi, merekam *track* dan upload *track* yang telah direkam. Pada Gambar 3.4 menunjukkan tampilan dari aplikasi Maverick.



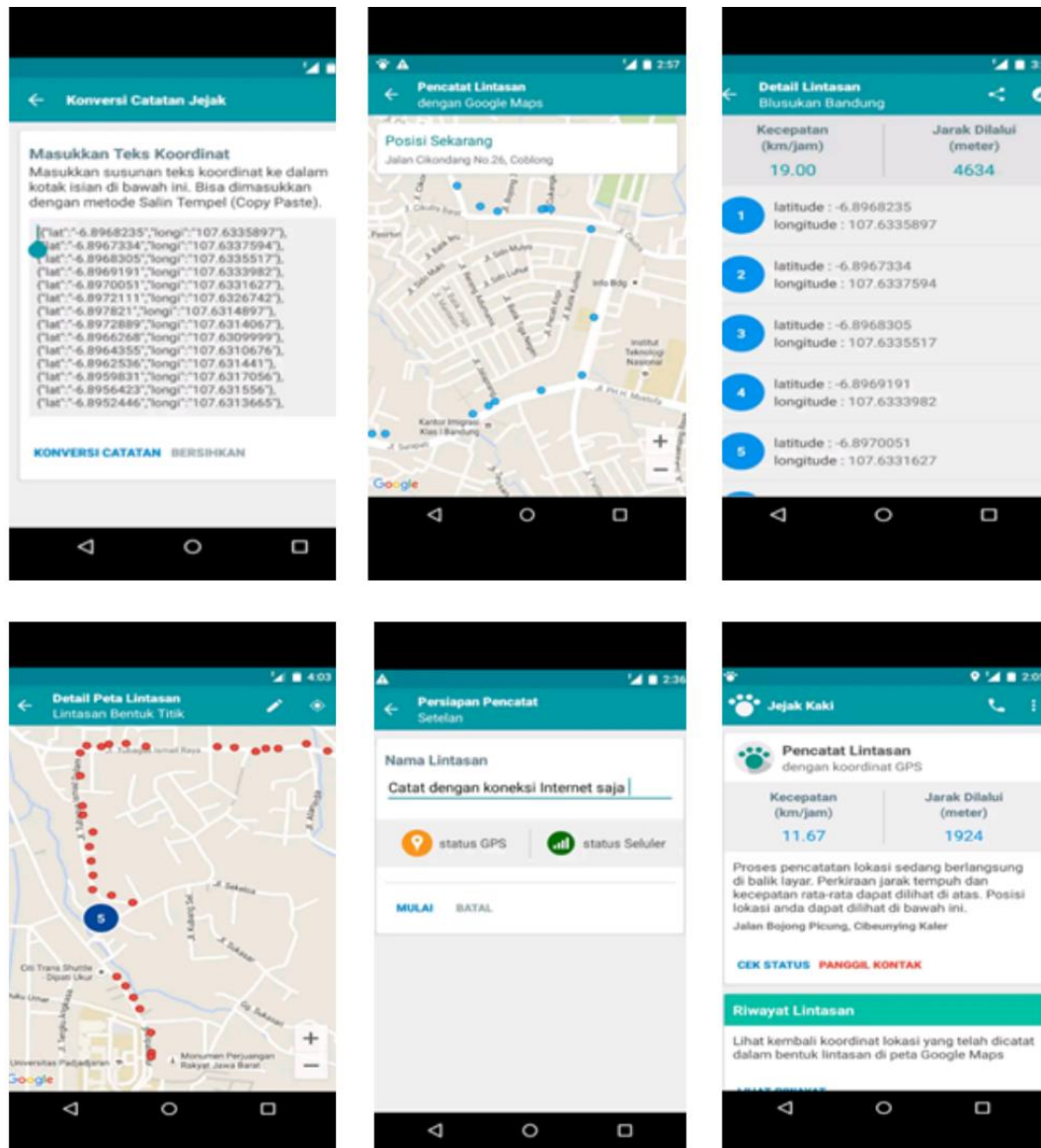


Gambar 3.4. Tampilan Maverick

### 3.2.5 Jejak Kaki

Aplikasi Jejak Kaki adalah aplikasi yang dapat digunakan untuk mencatat titik koordinat dari lokasi yang telah dilalui dan dapat menyimpannya dalam bentuk *latitude* dan *longitude*. Dengan aplikasi ini, pengguna dapat mencatat koordinat lokasi dari tempat atau jalan yang telah dilewatinya dan menyimpannya secara berurutan dari koordinat lokasi paling awal sampai koordinat lokasi paling akhir. Aplikasi ini akan menampilkan gambaran titik-titik koordinat lokasi tersebut ke dalam tampilan peta *online* sehingga pengguna dapat melihat kembali lokasi atau lintasan yang telah dilalui. Pengguna juga dapat melihat riwayat koordinat lokasi dalam bentuk lintasan jalur atau titik-titik lokasi GPS di dalam peta *online* yang disediakan di oleh aplikasi. Untuk fungsi aplikasi sama seperti aplikasi *tracking* lainnya yaitu menunjukkan lokasi keberadaan pengguna dan menunjukkan peta. Pada Gambar 3.5 menunjukkan tampilan dari aplikasi Jejak Kaki.



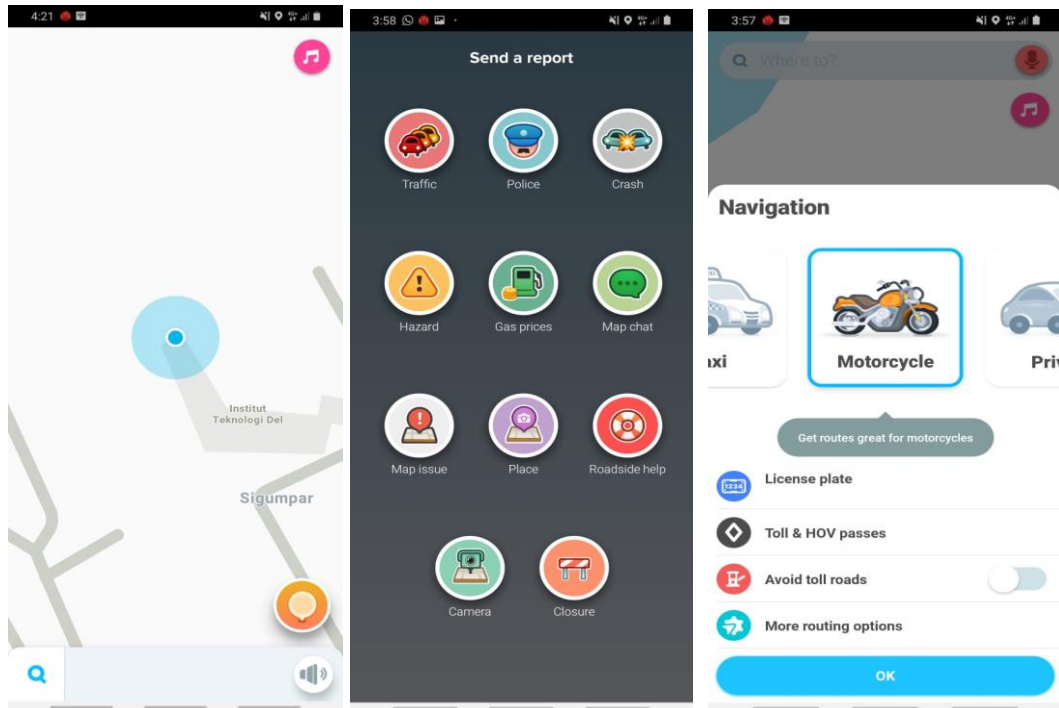


Gambar 3.5. Tampilan Jejak Kaki

### 3.2.6 Waze

Waze adalah sebuah aplikasi bertemakan navigasi dan lalu lintas terbesar di dunia. Waze dapat digunakan khususnya untuk melakukan pemantauan lokasi atau titik kemacetan dan mencari jalur alternatif yang tidak mengalami macet. Pada aplikasi ini, pengguna juga dapat mengetahui laporan kecelakaan, bahaya, razia polisi, dan penutupan jalan. Pengguna Waze lainnya dapat berbagi laporan lapangan yang memungkinkan pengguna untuk memantau kejadian-kejadian tidak terduga seperti kecelakaan di sepanjang rute perjalanan pengguna. Pengguna juga akan mendapat

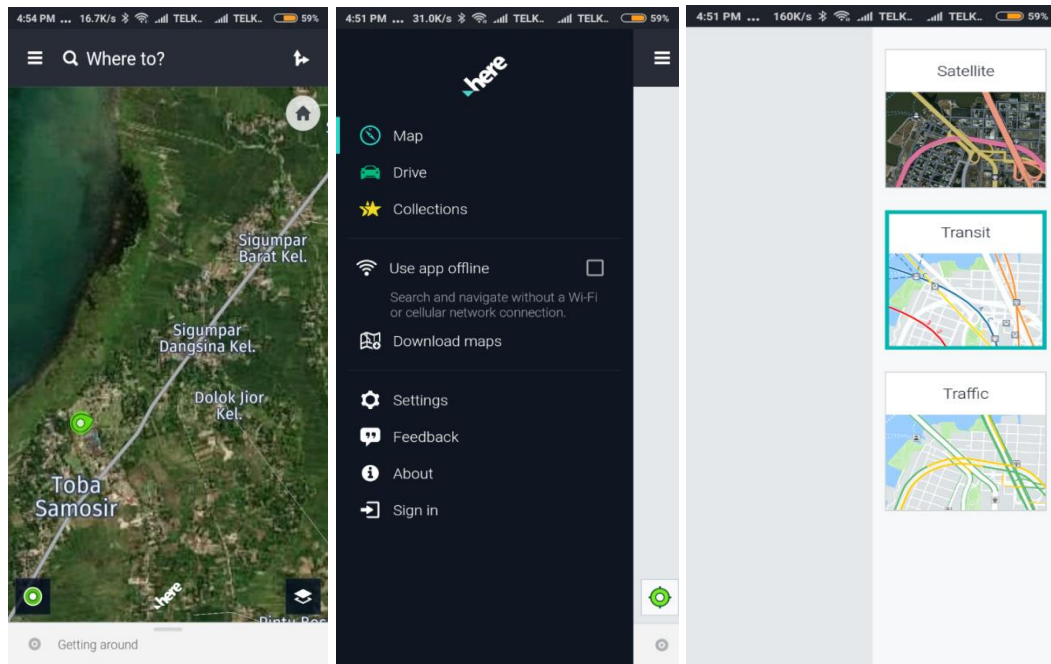
peta yang senantiasa disunting. Untuk fungsi aplikasi sama seperti aplikasi *tracking* lainnya yaitu menunjukkan lokasi keberadaan pengguna dan menunjukkan peta. Pada Gambar 3.6 menunjukkan tampilan dari aplikasi Waze.



**Gambar 3.6. Tampilan Waze**

### 3.2.7 Here WeGo

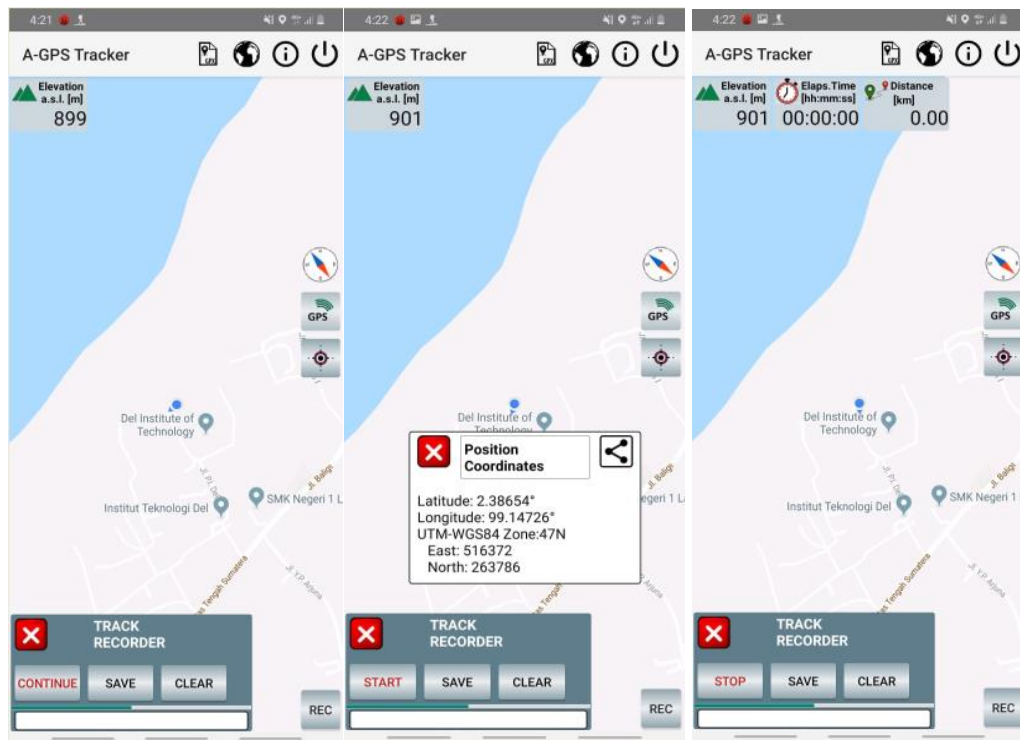
HERE WeGo adalah aplikasi gratis yang memudahkan navigasi kota, baik saat pengguna memerlukan taksi, transportasi umum atau membawa kendaraan sendiri. Pesan taksi dalam sekali ketuk atau mengemudi dengan rute terperinci, panduan belokan demi belokan, dan informasi mengenai semua cara yang dapat dilakukan untuk mencapai destinasi. Pada aplikasi ini juga akan menyediakan informasi transportasi umum di 1.300 kota. Untuk fungsi aplikasi sama seperti aplikasi *tracking* lainnya yaitu menunjukkan lokasi keberadaan pengguna dan menunjukkan peta. Pada Gambar 3.7 menunjukkan tampilan dari aplikasi Here WeGo.



Gambar 3.7. Tampilan HereWeGo

### 3.2.8 A-GPS Tracker

A-GPS Tracker adalah pelacak luar ruangan menggunakan kemampuan A-GPS telepon pengguna. Jika diaktifkan, pengguna dapat merekam lintasan pengguna bahkan ketika telepon masuk. Aplikasi ini dirancang khusus untuk pejalan kaki yang ingin merekam jalur baru atau ingin mengikuti jalur yang ada dan yang lebih penting ingin menghindari kehilangan pada saat perjalanan. Adapun fitur dari aplikasi adalah elevasi diberikan relatif terhadap permukaan laut rata-rata, koordinat Lintang dan Bujur pengguna diberikan dalam derajat dan di UTM-WSG84, trek dapat direkam, disimpan dalam file GPX atau dimuat ke atau dari memori dan dibagikan dengan orang lain, profil elevasi dan jaraknya dapat ditampilkan di layar, fungsi monitor "*follow path*" dapat diaktifkan untuk memberikan alarm jika Anda meninggalkan jalur GPX yang dimuat. Untuk fungsi aplikasi sama seperti aplikasi *tracking* lainnya yaitu menunjukkan lokasi keberadaan pengguna dan menunjukkan peta. Pada Gambar 3.8 menunjukkan tampilan dari aplikasi A-GPS Tracker.



Gambar 3.8. Tampilan A-GPS Tracker

Dari data perbandingan seluruh aplikasi diatas, dapat diperoleh perbandingan fitur-fitur yang terdapat pada aplikasi *tracking* yang umumnya digunakan pada Tabel 3.1.

**Tabel 3.1. Analisis Perbandingan Aplikasi Sejenis**

Fungsi	Aplikasi								Diimplemen- tasikan
	PolarSteps	Rute Saya	Maps	Maverick	Jejak Kaki	Waze	Here WeGo	A-GPS Tracker	
Mencari orang atau tempat	Tidak Tersedia	Tersedia	Tersedia	Tersedia	Tidak Tersedia	Tersedia	Tersedia	Tersedia	Tidak
Menambahkan sebuah rencana perjalanan	Tersedia	Tidak Tersedia	Tidak Tersedia	Tidak Tersedia	Tidak Tersedia	Tersedia	Tidak Tersedia		Tidak
Menunjukkan lokasi atau titik koordinat	Tidak Tersedia	Tersedia	Tersedia	Tersedia	Tidak Tersedia	Tersedia	Tersedia	Tersedia	Ya
Merekam rute perjalanan	Tersedia	Tersedia	Tersedia	Tidak Tersedia	Tersedia	Tidak Tersedia	Tidak Tersedia	Tersedia	Ya
Menggabungkan rute dengan foto	Tersedia	Tidak Tersedia	Tersedia	Tidak Tersedia	Tidak Tersedia	Tersedia	Tidak Tersedia	Tidak Tersedia	Ya
Membagikan atau mengimpor rute dari ponsel atau Google Drive	Tersedia	Tersedia	Tersedia	Tidak Tersedia	Tersedia	Tidak Tersedia	Tersedia	Tersedia	Ya
Navigasi berpadukan suara	Tidak Tersedia	Tidak Tersedia	Tersedia	Tidak Tersedia	Tidak Tersedia	Tersedia	Tidak Tersedia	Tidak Tersedia	Ya

Terhadap hasil analisis dari pengamatan langsung pada aplikasi *tracking* yang umum dipakai oleh *user*, secara umum kebutuhan aplikasi *tracking* yang dibutuhkan seperti berikut :

1. Menampilkan lokasi titik koordinat

Peta merupakan fitur utama yang menampilkan dimana pengguna berada sekarang. Peta akan menampilkan lokasi dan titik koordinat pengguna berada. Peta pada aplikasi Track My Trip merupakan halaman pertama yang akan ditampilkan kepada *user* saat membuka halaman aplikasi Track My Trip.

2. Membagikan perjalanan

Membagikan rute perjalanan yang telah direkam dapat dibagikan oleh pengguna kepada pengguna lain seperti teman, keluarga, dll. Rute perjalanan tersebut dapat dibagikan melalui WhatsApp, Facebook, maupun sosial media lainnya.

3. Merekam rute perjalanan

Fitur merekam rute perjalanan dapat digunakan pengguna untuk merekam titik atau rute perjalanan yang telah dilalui oleh pengguna.

4. Mencari lokasi

Fitur mencari lokasi dapat digunakan pengguna untuk mencari atau menemukan lokasi yang ingin dijalani atau dikunjungi.

Selain fitur-fitur di atas, berikut beberapa fitur utama yang dibutuhkan aplikasi *tracking* yang akan dikembangkan bukan hasil analisis dari aplikasi sejenis, yaitu aplikasi yang dapat merelasikan lokasi dengan beberapa multimedia yaitu foto, video, audio, dan note.

1. Merelasikan foto dengan lokasi

Fitur ini dapat digunakan pengguna untuk merelasikan foto yang telah diambil dengan lokasi pengambilan foto.

2. Merelasikan video dengan lokasi

Fitur ini dapat digunakan pengguna untuk merelasikan video yang telah diambil dengan lokasi pengambilan video.

3. Merelasikan audio dengan lokasi

Fitur ini dapat digunakan pengguna untuk merelasikan audio yang telah diambil dengan lokasi pengambilan audio.

#### 4. Merelasikan note dengan lokasi

Fitur ini dapat digunakan pengguna untuk merelasikan note yang telah diambil dengan lokasi pengambilan note.

### 3.3 Analisis GPS (*Global Positioning System*)

GPS akan menyediakan bentuk data berupa titik-titik koordinat *latitude* dan *longitude* dari lokasi keberadaan seseorang. GPS dalam penerapannya sangat umum dan beragam karena tidak terfokus pada satu *platform* tertentu namun dapat diterapkan dalam berbagai *platform* tergantung pengembang dan penggunaannya, beberapa contoh penerapan GPS antara lain, penerapan GPS dalam kendaraan yang berguna sebagai navigator perjalanan, dalam dunia militer GPS juga digunakan sebagai alat navigasi untuk melakukan pelacakan dan navigasi tempat atau area tertentu, penggunaan GPS dalam aplikasi *mobile* seperti GRAB, GO-JEK, dan aplikasi sejenis lainnya dalam hal ini GPS digunakan untuk mengetahui area atau lokasi kerja seorang driver dan clientnya yang melakukan order, sehingga dengan demikian driver dapat melakukan pengantaran atau penjemputan penumpang dari suatu tempat atau ke tempat lainnya dengan akurat. Pada pengembangan aplikasi ini akan menggunakan GPS untuk menunjukkan titik koordinat dari pengguna, mengambil titik koordinat dari seseorang, dan menentukan waktu perjalanan dari pengguna. Berdasarkan 5 layanan yang ada pada GPS, 3 diantaranya akan dipakai untuk pengembangan aplikasi ini diantaranya :

#### 1. Lokasi

Layanan lokasi pada GPS digunakan pada aplikasi ini untuk menunjukkan titik koordinat posisi maupun lokasi dari pengguna.

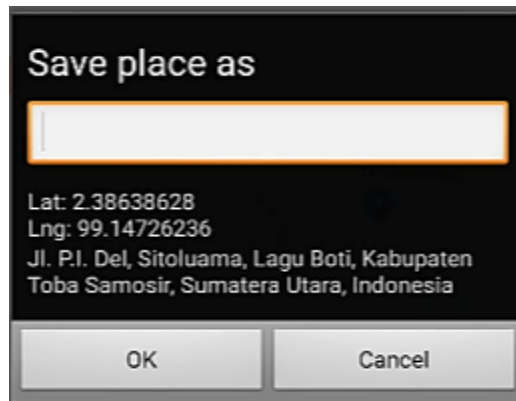
#### 2. *Tracking*

*Tracking* pada aplikasi ini digunakan untuk memonitoring perjalanan atau pengambilan titik yang akan dilalui oleh pengguna.

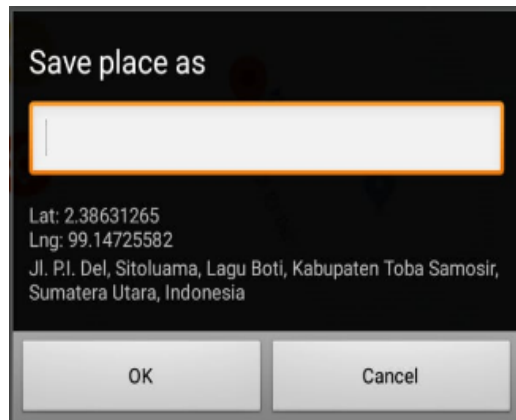
#### 3. *Timing*

*Timing* digunakan pada aplikasi ini digunakan untuk menentukan waktu atau lamanya perjalanan dari pengguna.

Pada Gambar 3.9 dan Gambar 3.10 adalah contoh perbandingan data pengambilan GPS yang pengambilan dilakukan menggunakan *device* yang berbeda dengan lokasi yang berdekatan. Data menunjukkan bahwa jarak antara *latitude* dan *longitude* tidak jauh berbeda karena perbedaan hanya terdapat di 4 angka terakhir.



**Gambar 3.9. Data Pengambilan GPS Menggunakan Device 1**

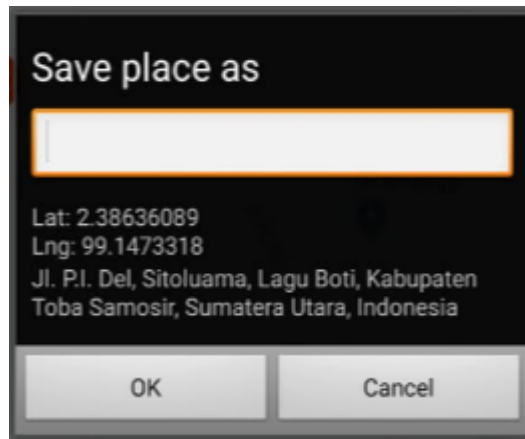


**Gambar 3.10. Data Pengambilan GPS Menggunakan Device 2**

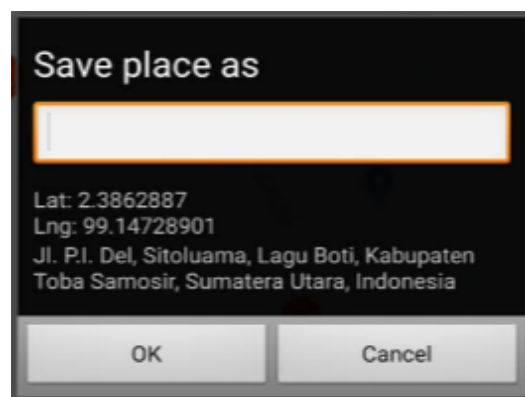
Pada Gambar 3.11 dan Gambar 3.12 adalah contoh perbandingan data pengambilan GPS yang pengambilan dilakukan menggunakan *device* yang sama dengan lokasi berbeda yaitu perbedaan jarak di radius 10 meter. Data menunjukkan bahwa jarak antara *latitude* dan *longitude* dengan 2 *device* mirip dengan pengambilan titik



kurang dari 10 meter. Tetapi, pengambilan titik lebih dari radius 10 menunjukkan perbedaan yaitu terdapat di 5 angka terakhir.



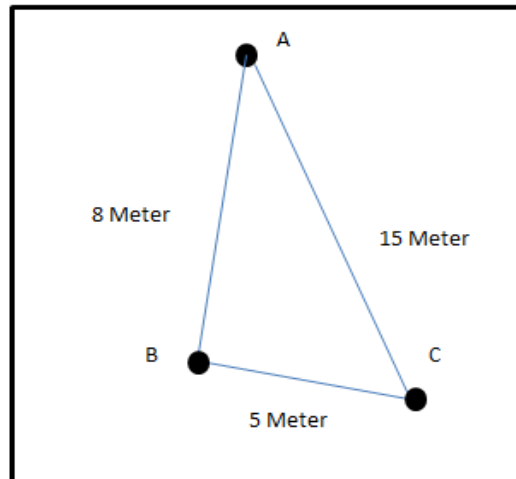
**Gambar 3.11. Data Pengambilan GPS Lebih Kecil 10 Meter**



**Gambar 3.12. Data Pengambilan GPS Lebih Besar 10 Meter**

Dalam pengembangan aplikasi ini, *tracking* akan dilakukan setiap 10 menit perjalanan. Misalnya pengguna melakukan perjalanan ke suatu tempat dengan lama perjalanan 30 menit dan terdapat 3 titik yang telah diambil. Di 10 menit pertama, pengguna mengambil foto dan telah melalui titik dari A ke B dengan radius 8 meter, maka foto akan tersimpan di titik terakhir pengguna mulai mengambil foto yaitu titik A. Di 10 menit kedua, pengguna mengambil foto lagi dan telah melalui titik dari A ke C dengan radius 15 meter, maka foto akan tersimpan di dua titik yaitu titik terakhir pengguna mulai mengambil foto yaitu titik A dan titik pengguna

mengakhiri pengambilan foto yaitu titik B. Dan untuk kondisi yang lain yaitu pada saat melakukan perekaman titik pengguna tidak berpindah tempat secara signifikan, sementara rekaman video/voice terus berjalan. Maka diasumsikan bahwa titik kedua tidak ada, maka hasil rekaman akan disimpan pada titik terakhir dimana rekaman video atau voice diambil. Pada Gambar 3.13 menunjukkan ilustrasi pengambilan titik koordinat.



**Gambar 3.13. Ilustrasi Pengambilan Titik Koordinat**

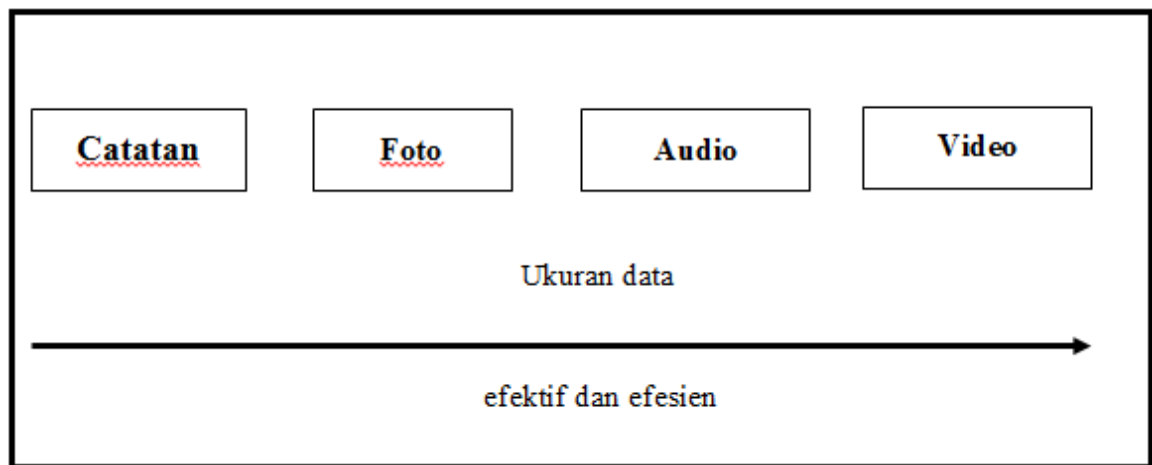
GPS merupakan layanan yang dapat di akses dengan adanya alat penerima layanan dari satellite GPS yang telah disematkan atau dipasang dalam suatu alat atau device, seperti smartphone, navigasi kendaraan dll. Dalam Tugas Akhir ini sensor layanan GPS pada smartphone dimanfaatkan untuk memperoleh data lokasi pengguna berupa titik koordinat secara realtime dan akurat. Untuk mengakses layanan GPS hanya dengan mengijinkan permission untuk GPS mengakses lokasi device, kemudian GPS akan memberikan informasi titik lokasi device saat itu juga. Pada Gambar 3.14 menunjukkan interaksi antara *user*, device dan GPS.



**Gambar 3.14. Interaksi antara User, Device dan GPS**

### 3.4 Analisis Penggunaan Multimedia

Aplikasi yang akan dikembangkan adalah suatu aplikasi *tracking* atau pengambilan titik yang dapat merelasikannya dengan berbagai multimedia. Multimedia akan bertujuan menyampaikan informasi lebih efektif dalam waktu yang terbatas. Multimedia akan merangsang indera penting manusia diantaranya penglihatan, pendengaran, gerak, ataupun suara. Foto, video, voice dan note adalah multimedia yang akan digunakan pada Tugas Akhir dengan tingkat penyampaian informasi yang lebih efektif. Pada Gambar 3.15 menjelaskan bagaimana tingkat penyampaian informasi data multimedia yang digunakan dan ukuran data yang berbeda. Semakin ke kanan maka data multimedia semakin efektif menyampaikan informasi.



Gambar 3.15. Analisis Data Multimedia

(Sumber : telah diolah kembali dari :

<http://catatansimbakpacker.blogspot.com/2016/11/pentingnya-multimedia-dalam-kehidupan.html> )

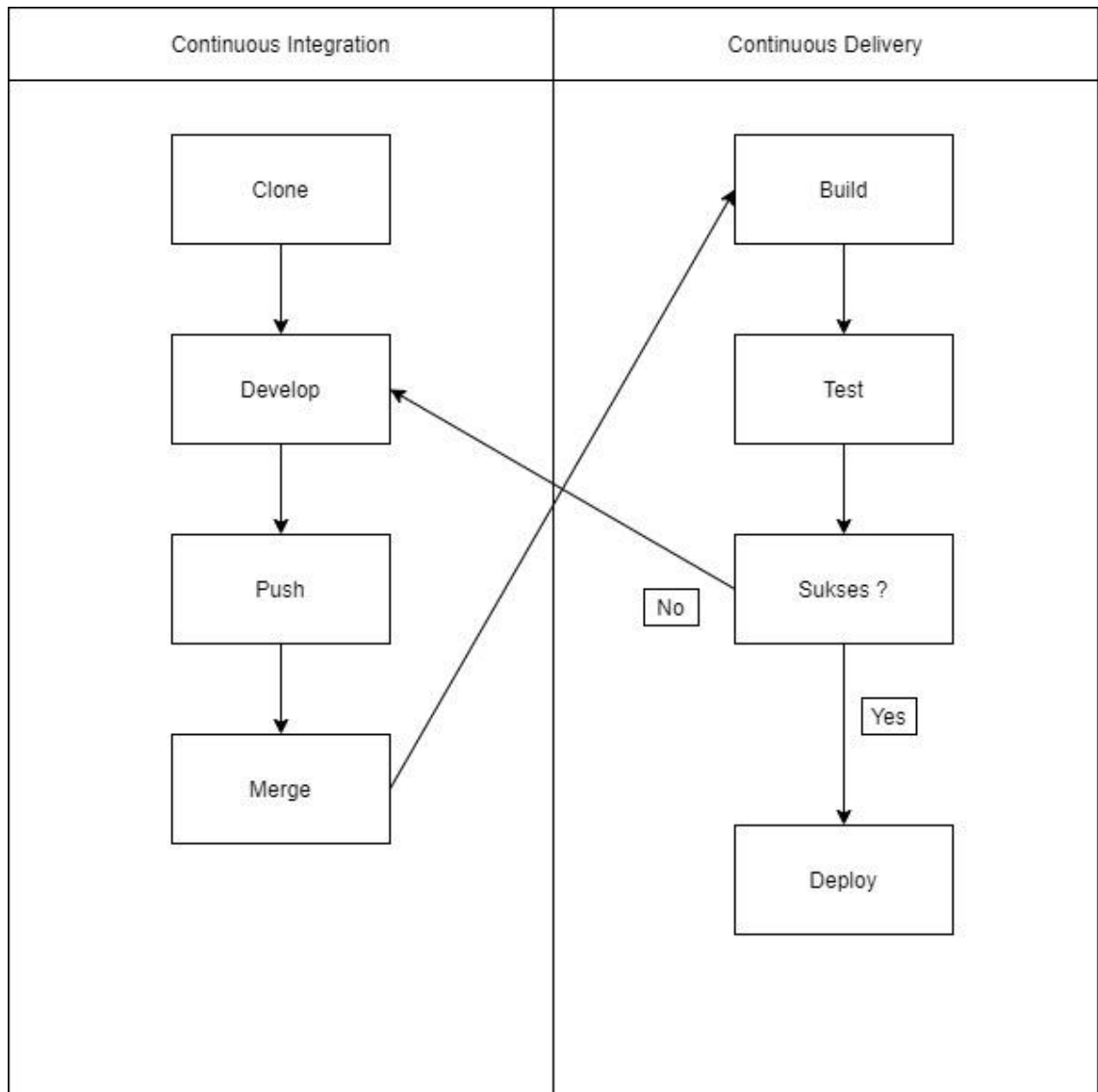
Catatan merupakan data dengan bobot ukuran data yang tidak besar, dalam aplikasi yang akan dikembangkan ini catatan dapat digunakan sebagai pengingat untuk perjalanan yang dilalui, catatan tempat yang menurut *user* perlu didokumentasikan. Foto bisa dimanfaatkan untuk mengabadikan perjalanan yang dilalui, foto dapat disimpan dalam aplikasi dan dapat mendeskripsikan apa yang ada dalam catatan perjalanan yang telah dibuat, namun dengan adanya catatan dan foto dalam aplikasi akan berguna sebagai data yang menjelaskan atau menggambarkan data perjalanan yang dilalui, namun tidak semua orang dapat memahaminya dengan baik sekalipun data perjalan telah disimpan dalam bentuk gambar dan catatan, maka hal ini dapat

lebih di akuratkan lagi dengan adanya perekaman audio yang bermanfaat sebagai data yang menjelaskan perjalanan dengan menceritakan perjalanan tersebut, menjelaskan data catatan dan foto yang telah tersimpan sebelumnya. Data yang berupa catatan, foto dan audio dapat mendeskripsikan tentang perjalanan namun harus terdapat banyak catatan, foto dan audio untuk dapat menjelaskan perjalanan dengan detail, maka dengan demikian video merupakan alternatif untuk menjadi pelengkap dari data tersebut, *user* dapat menjelaskan perjalanannya dengan cara merekam video perjalanan, sambil melakukan perekaman perjalanan *user* dapat menjelaskan perjalanan sambil menunjukan pemandangan ataupun menunjukan gambaran daerah sekitar tempat *user* berada.

### 3.5 Analisis CICD dengan Tools Jenkins

Untuk penerapan CICD pada Tugas Akhir ini akan diterapkan pada saat implementasi dimana masing-masing developer mengerjakan satu fungsi dalam satu aplikasi secara paralel. Tools yang akan digunakan adalah Jenkins tools berbasis *automation* yang memiliki fitur antara lain *build*, mengambil *code* dari *repository*, *build* dan *test* secara otomatis dan menghasilkan laporan hasil *build*. Untuk *repository* yang digunakan pada pengembangan aplikasi ini adalah GitLab. *Developer* akan menyatukan kode yang telah dikerjakan dengan cara melakukan *push* ke dalam *repository* yang telah dibuat. Jika proses *push* dari *developer* berhasil atau tidak error, maka akan lanjut ke proses *build* Jenkins.

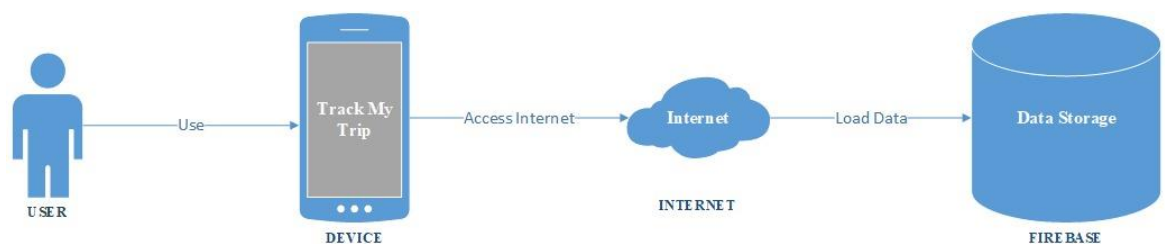
Pada Gambar 3.16 akan menggambarkan proses CICD.



**Gambar 3.16. Analisis Proses CICD**

### 3.6 Desain Arsitektur Aplikasi

Berikut pada Gambar 3.17 desain arsitektur sistem aplikasi Track My Trip yang akan digunakan selama pengerjaan Tugas Akhir.



**Gambar 3.17. Arsitektur Aplikasi**

**Keterangan :**

1. User mengakses dan mengaktifkan aplikasi Track My Trip melalui smartphone dan menggunakan koneksi internet.
2. Device terkoneksi ke internet untuk melakukan load dan manipulasi data dari Firebase.
3. *Firebase* berguna sebagai database dari aplikasi. *Firebase* bertugas untuk menyimpan seluruh data yang disimpan oleh user. *Firebase* digunakan untuk mempermudah proses pengembangan aplikasi.

**3.7 Analisis Kebutuhan**

Berdasarkan analisis yang dilakukan pada sub bab analisis GPS, analisis penggabungan titik koordinat dengan *multiple data*, penulis mendapatkan bahwa kebutuhan akan sebuah aplikasi *tracking* perlu ditingkatkan. Kebutuhan tersebut adalah perekaman atau pendokumentasian suatu rute perjalanan dalam data baik itu titik koordinat, foto, video, note, dan audio dan merelasikannya dengan waktu dan tempat secara berkala. Aplikasi tersebut akan diberi nama *Track My Trip*. Aplikasi dikembangkan dengan versi *mobile* yang dapat digunakan pengguna untuk menunjukkan lokasi pengguna berada, merekam perjalanan, merelasikan lokasi dengan foto, merelasikan lokasi dengan video, merelasikan lokasi dengan note, dan merelasikan lokasi dengan *voice record*.

**3.7.1 Functional Requirement**

Pada Tabel 3.2 menjelaskan mengenai kebutuhan fungsional (*functional requirement*) dari aplikasi yang akan dikembangkan.

**Tabel 3.2. Functional Requirement**

No.	Nama Fungsi	Main Function	Deskripsi
1.	FR-01	Melakukan autentikasi	User dapat mendaftarkan akun nya dan masuk ke dalam aplikasi
2.	FR-02	Melihat lokasi pengguna	User dapat melihat lokasi keberadaannya

No.	Nama Fungsi	Main Function	Deskripsi
3.	FR-03	Melakukan perekaman perjalanan	<i>User</i> dapat melakukan perekaman titik koordinat perjalanan yang telah dilalui
4.	FR-04	Menghapus rekaman perjalanan	<i>User</i> dapat menghapus data rekaman perjalanan
5.	FR-05	Merelasikan titik koordinat dengan foto	<i>User</i> dapat merelasikan foto dengan titik koordinat yang telah direkam
6.	FR-06	Merelasikan titik koordinat dengan video	<i>User</i> dapat merelasikan video dengan titik koordinat yang telah direkam
7.	FR-07	Merelasikan titik koordinat dengan voice record	<i>User</i> dapat merelasikan voice record dengan titik koordinat yang telah direkam
8.	FR-08	Merelasikan titik koordinat dengan note	<i>User</i> dapat merelasikan note dengan titik koordinat yang telah direkam
9.	FR-09	Share perjalanan	<i>User</i> dapat membagikan data perjalanan yang telah direkam
10.	FR-10	Impor data perjalanan	<i>User</i> dapat mengimpor data perjalanan

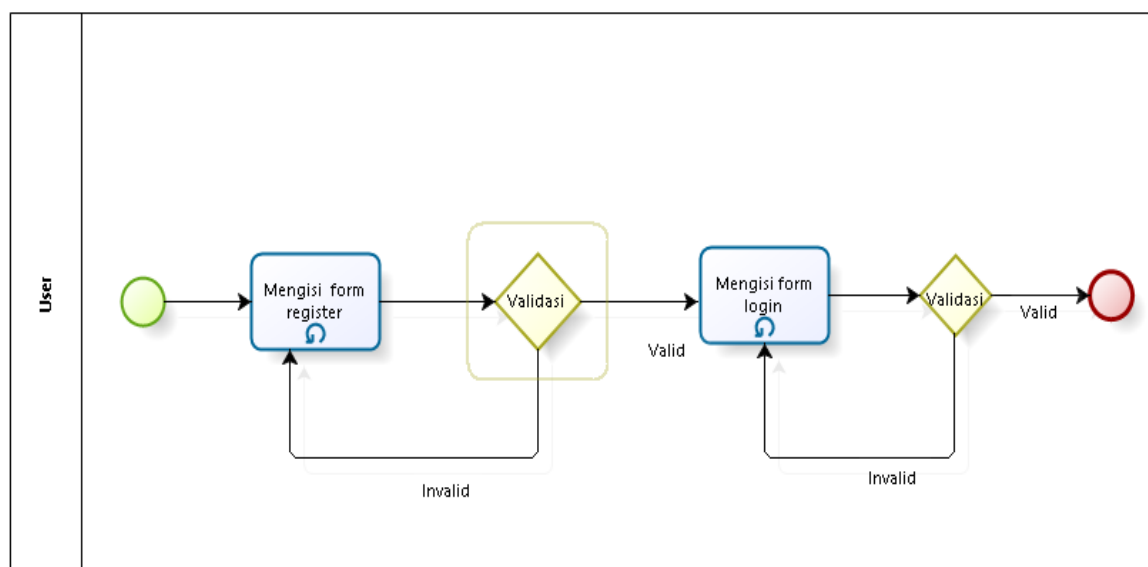
### 3.7.2 Business Process

Pada bagian ini dijelaskan mengenai proses bisnis yang terjadi pada perancangan target sistem aplikasi Track My Trip.

#### 3.7.2.1 Melakukan Autentikasi

Fungsi ini akan digunakan pengguna untuk mendaftarkan akunnya ke dalam aplikasi dan login sebelum masuk ke dalam aplikasi. Pengguna terlebih dahulu membuka aplikasi dan masuk ke halaman register. Setelah masuk ke halaman register, pengguna akan mengisi form register dengan memasukkan nama lengkap,

alamat email, nomor hp, *password*, dan konfirmasi *password*. Ketika pengguna salah mengisi form register maka pengguna harus kembali ke proses awal yaitu mengisi form register dengan memasukkan nama lengkap, alamat email, nomor hp, *password* dan konfirmasi *password*. Tetapi ketika pengguna benar mengisi form register maka pengguna akan masuk ke halaman login. *Username* dan *password* yang telah didaftarkan harus dimasukkan pengguna terlebih dahulu. Akan ada proses validasi *username* dan *password*. Ketika pengguna memasukkan *username* atau *password* yang salah maka pengguna harus kembali ke proses awal yaitu memasukkan *username* dan *password*. Pada Gambar 3.18 berikut ini proses untuk melakukan registrasi.

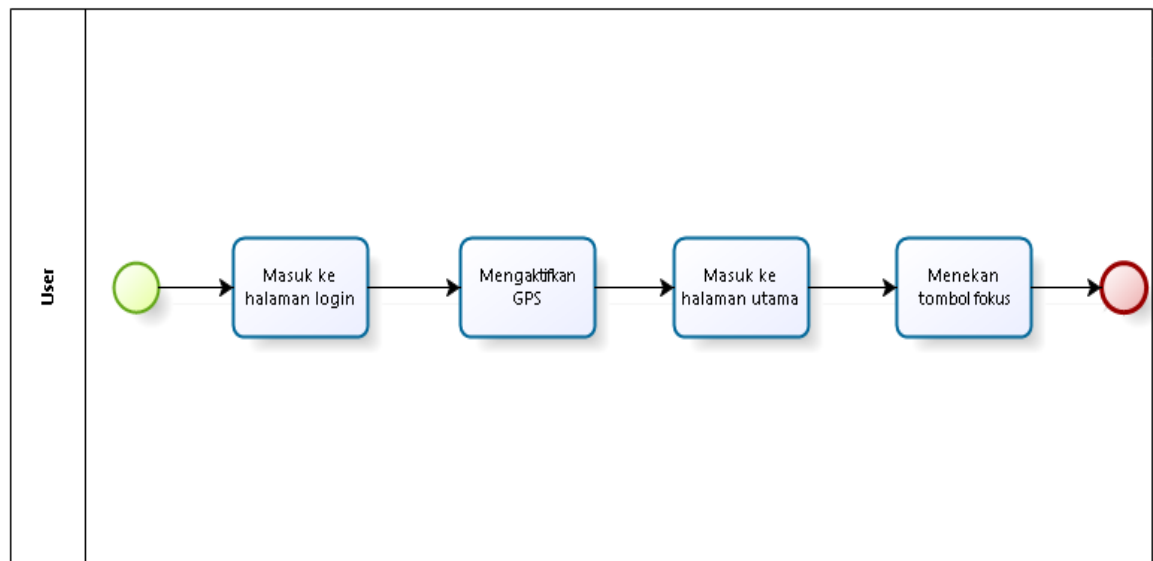


**Gambar 3.18. Business Process Melakukan Autentikasi**

### 3.7.2.2 Menunjukkan Lokasi Pengguna

Fungsi ini akan digunakan pengguna untuk menunjukkan lokasi pengguna dimana pengguna berada. Pengguna masuk ke halaman home terlebih dahulu dan melihat peta yang ada pada aplikasi. Dengan mengaktifkan fitur GPS, maka aplikasi akan menunjukkan peta. Untuk melihat keberadaan pengguna, maka pengguna harus menekan tombol fokus. Pada Gambar 3.19 berikut ini proses untuk menunjukkan lokasi pengguna.

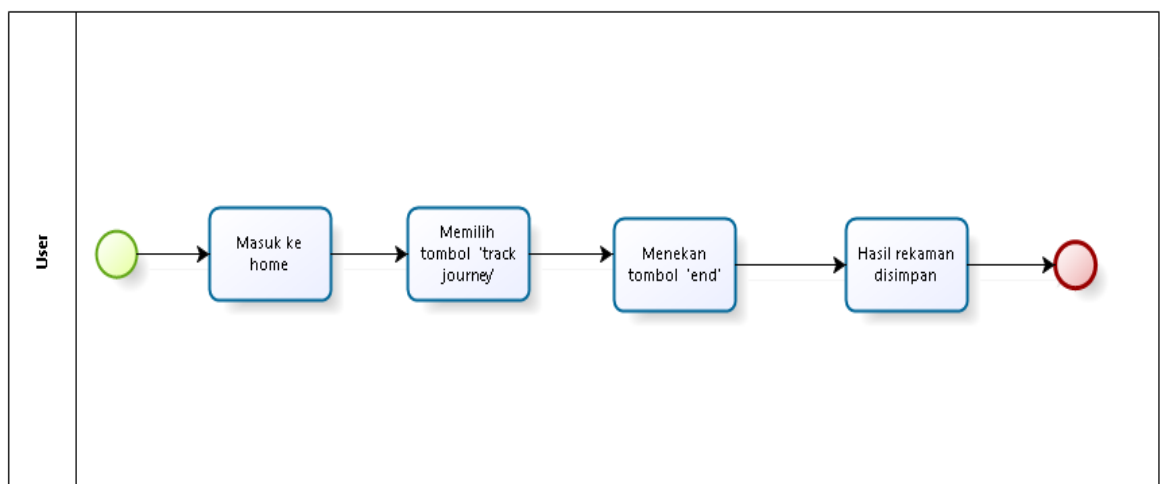




**Gambar 3.19. Business Process Menampilkan Lokasi**

### 3.7.2.3 Rekam Perjalanan

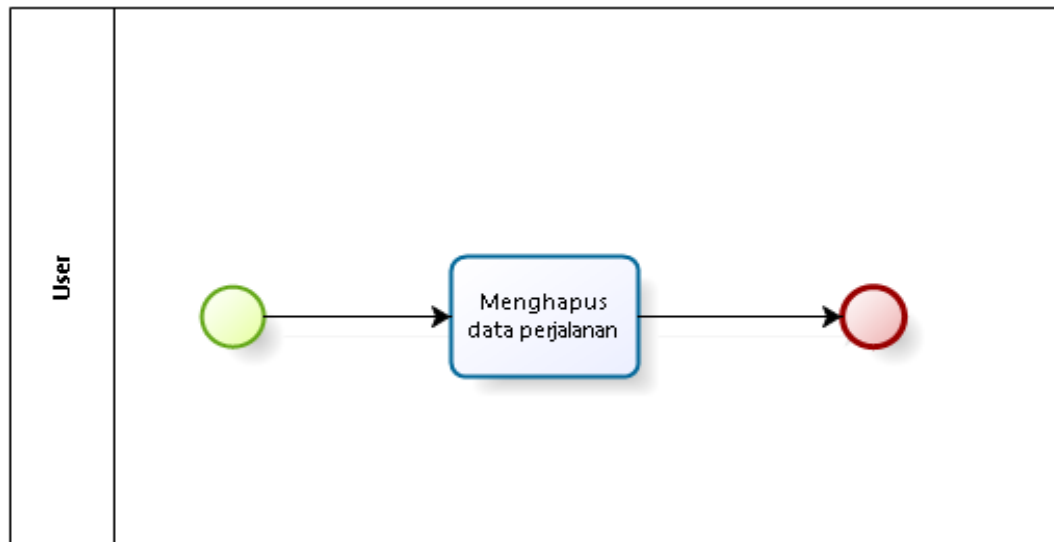
Fungsi ini akan digunakan pengguna untuk merekam perjalanan. Pengguna dapat merekam perjalanan yang akan akan dituju. Pengguna terlebih dahulu menekan *button track journey* pada aplikasi. Perekaman akan dilakukan setiap 10 menit dan jika perekaman telah selesai maka pengguna menekan tombol *end*. Pada Gambar 3.20 ini menunjukkan bisnis proses untuk melakukan rekam perjalanan.



**Gambar 3.20. Business Process Rekam Perjalanan**

### 3.7.2.4 Menghapus Data Perjalanan

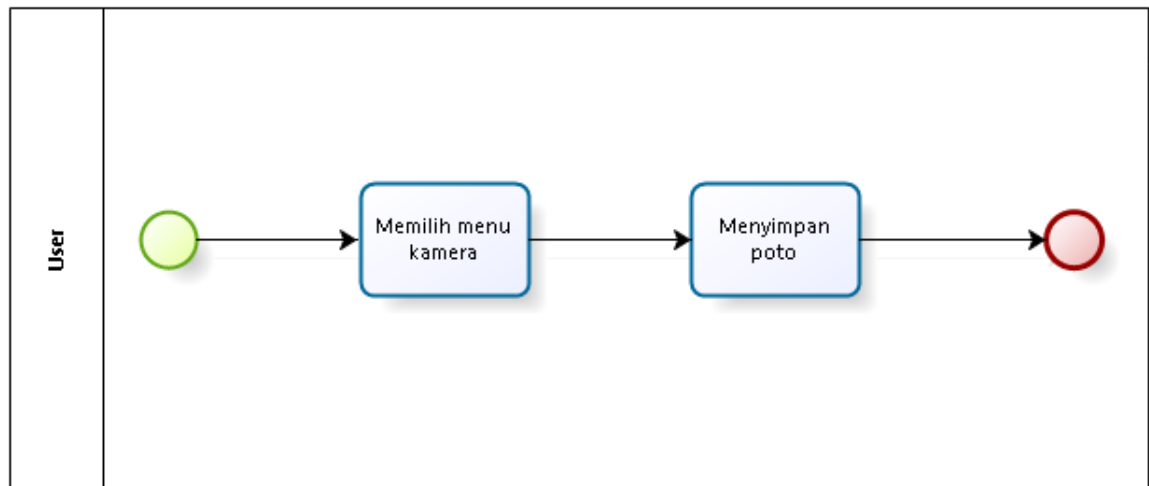
Fungsi ini akan digunakan pengguna untuk menghapus data perjalanan yang telah direkamnya. Pengguna telah masuk ke dalam aplikasi, dan melihat daftar perjalanan yang telah direkam. Pada Gambar 3.21 ini menunjukkan bisnis proses untuk menghapus data perjalanan.



**Gambar 3.21. Business Process Menghapus Perjalanan**

### 3.7.2.5 Merelasikan Lokasi dengan Foto

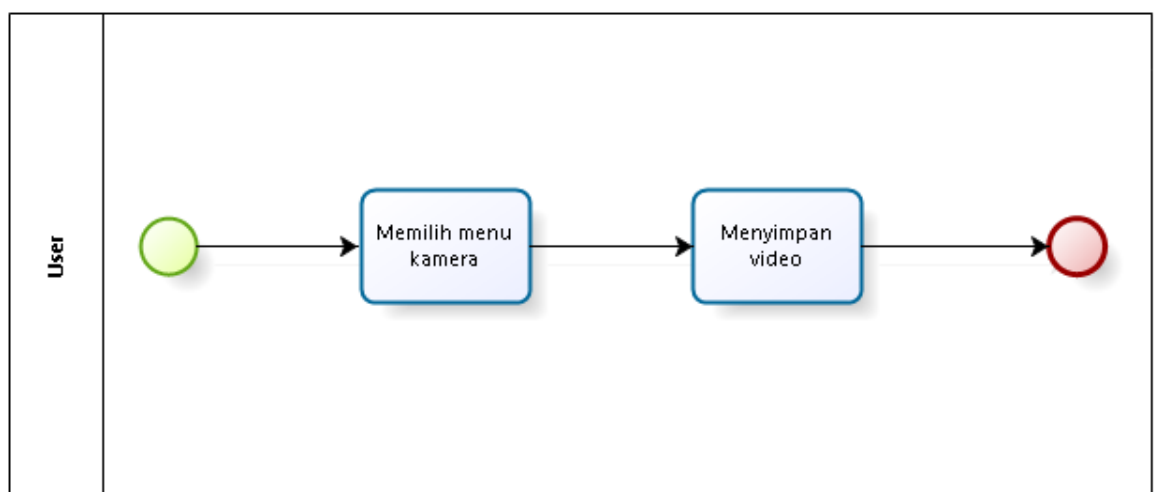
Fungsi ini akan digunakan pengguna untuk merelasikan foto yang diambilnya dengan lokasi titik tempat pengguna mengambil foto nya. Pengambilan foto hanya dapat dilakukan pengguna jika terlebih dahulu perekaman perjalanan telah diaktifkan. Memilih menu yaitu kamera. Maka aplikasi akan mulai mengaktifkan kamera dan foto akan diambil. Setelah foto diambil, pengguna akan bisa langsung menyimpan foto atau tidak menyimpan foto tersebut. Pada Gambar 3.22 ini menunjukkan bisnis proses untuk melakukan merelasikan lokasi dengan foto.



**Gambar 3.22. Business Process Merelasikan Lokasi dengan Foto**

#### **3.7.2.6 Merelasikan Lokasi dengan Video**

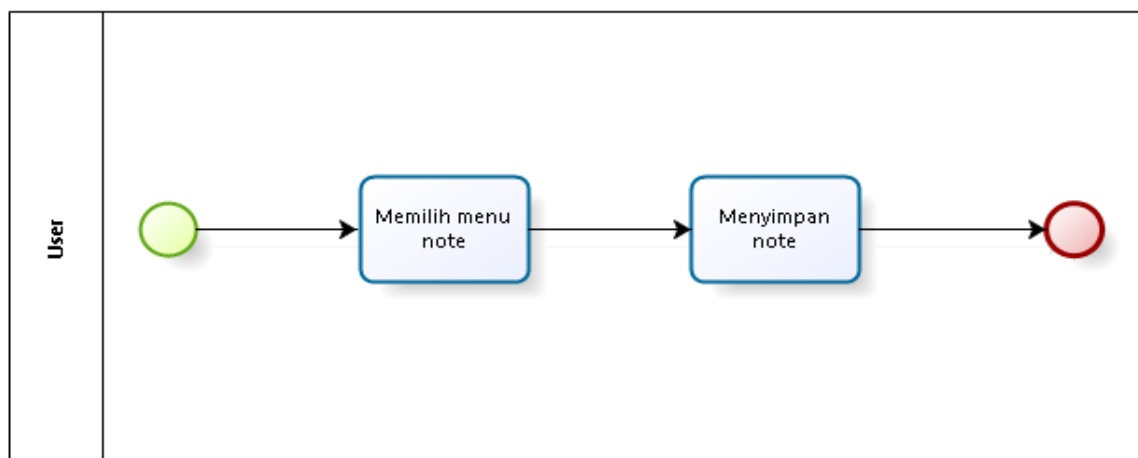
Fungsi ini akan digunakan pengguna untuk merelasikan sebuah video yang diambilnya dengan lokasi titik tempat pengguna merekam video. Pengambilan video hanya dapat dilakukan pengguna jika terlebih dahulu perekaman perjalanan telah diaktifkan. Memilih menu kamera. Maka aplikasi akan mulai merekam video saat perjalanan. Setelah video diambil, pengguna akan bisa langsung menyimpan video atau tidak menyimpan video tersebut. Pada Gambar 3.23 ini menunjukkan bisnis proses untuk melakukan merelasikan lokasi dengan video.



**Gambar 3.23. Business Process Merelasikan Lokasi dengan Video**

### 3.7.2.7 Merelasikan Lokasi dengan Note

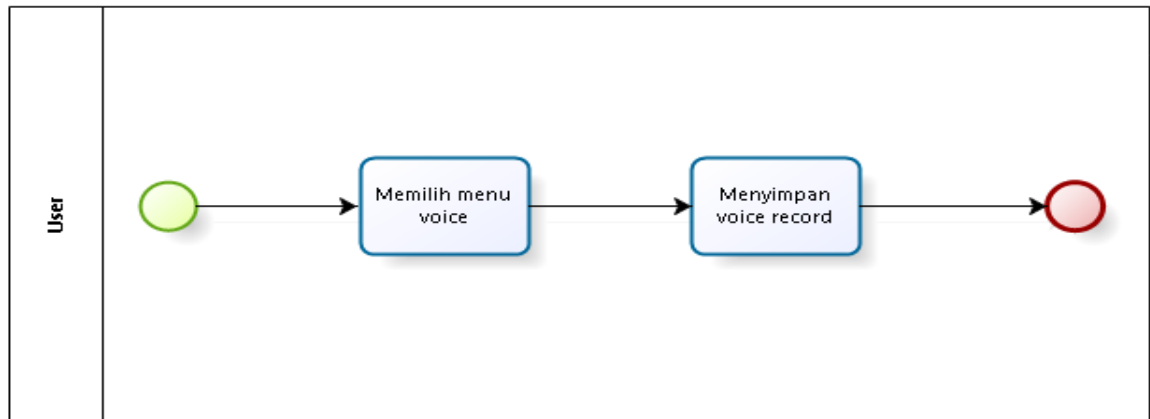
Fungsi ini akan digunakan pengguna untuk menulis sebuah catatan dalam bentuk teks. Pengambilan note hanya dapat dilakukan pengguna jika terlebih dahulu perekaman perjalanan telah diaktifkan. Memilih menu note maka pengguna dapat menulis sebuah catatan pada aplikasi. Setelah note dicatat, pengguna akan bisa langsung menyimpan note atau tidak menyimpan note tersebut. Pada Gambar 3.24 ini menunjukkan bisnis proses untuk melakukan merelasikan lokasi dengan note.



Gambar 3.24. Business Process Merelasikan Lokasi dengan Note

### 3.7.2.8 Merelasikan Lokasi dengan voice record

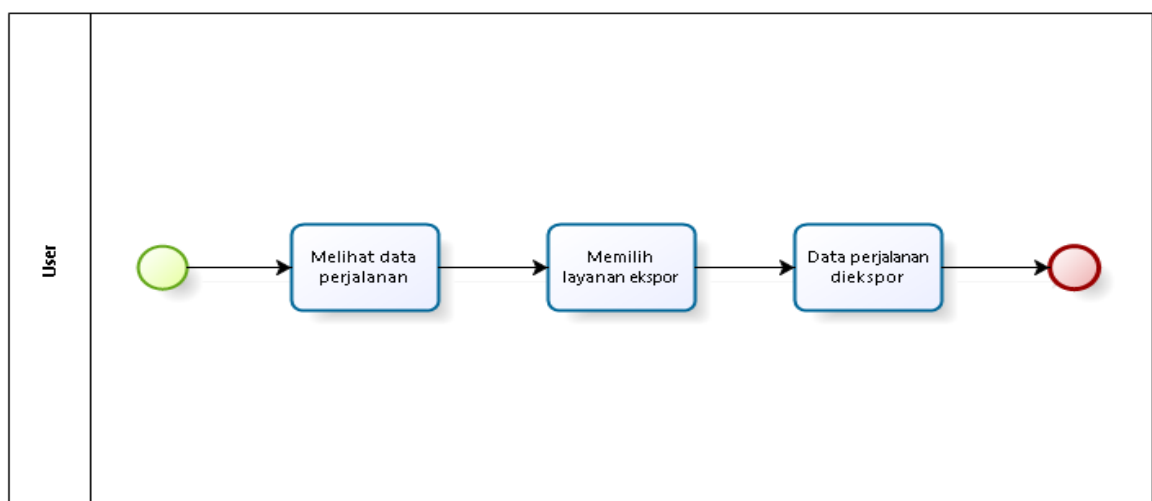
Fungsi ini akan digunakan pengguna untuk merekam suara pengguna ketika ingin memulai maupun memberhentikan aplikasi. Pengambilan voice hanya dapat dilakukan pengguna jika terlebih dahulu perekaman perjalanan telah diaktifkan. Memilih menu voice, maka pengguna dapat merekam suara. Setelah voice direkam, pengguna akan bisa langsung menyimpan voice atau tidak menyimpan voice tersebut. Pada Gambar 3.25 ini menunjukkan bisnis proses untuk melakukan merelasikan lokasi dengan voice record.



**Gambar 3.25. Business Process Merelasikan Lokasi dengan Voice**

### 3.7.2.9 Share Perjalanan

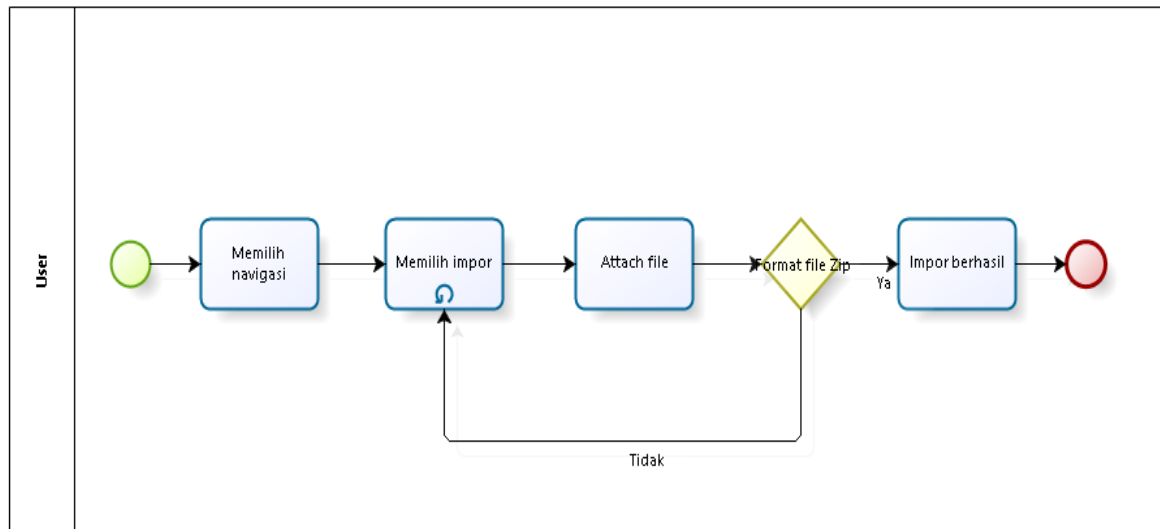
Fungsi ini akan digunakan pengguna untuk membagikan data perjalanan yang telah direkam kepada pengguna lain. Pada Gambar 3.26 ini menunjukkan bisnis proses untuk share perjalanan.



**Gambar 3.26. Business Process Share perjalanan**

### 3.7.2.10 Impor Data Perjalanan

Fungsi ini akan digunakan pengguna untuk mengimpor data perjalanan yang telah direkam. Pada Gambar 3.27 ini menunjukkan bisnis proses untuk impor data perjalanan.



**Gambar 3.27. Business Process Impor Data Perjalanan**

### 3.7.3 Non-Functional Requirement

Pada bagian ini menjelaskan kebutuhan *non-functional requirement* aplikasi yang mendukung berjalannya aplikasi agar aplikasi dapat berjalan sesuai dengan kebutuhan pengguna. Pada Tabel 3.3 dapat dilihat kebutuhan *non-functional requirement* dari aplikasi yang akan dikembangkan.

**Tabel 3.3. Non-Functional Requirement**

No	Non-Functional Requirement	Parameter	Requirement
1.	NFR-01	<i>Usability</i>	Aplikasi yang dibangun mudah untuk digunakan oleh pengguna dan sesuai dengan kebutuhan pengguna
2.	NFR-02	<i>Flexisibility</i>	Aplikasi yang digunakan mudah untuk disesuaikan dengan semua pengguna

### 3.8 Desain

Pada bagian ini dijelaskan mengenai rancangan aplikasi yang dikembangkan yaitu perancangan aplikasi yang terdiri mulai dari *use case diagram*, *conceptual data*

*model, physical data model, class diagram, dan sequence diagram* dan perancangan kebutuhan data aplikasi pada aplikasi.

### 3.8.1 User Characteristic

Pada bagian ini menjelaskan karakteristik dari pengguna aplikasi versi *mobile*.

#### 3.8.1.1. User Group 1

<i>Description of user</i>	: <i>User</i>
<i>Role</i>	: <i>User</i> dapat menggunakan aplikasi versi <i>mobile</i> .
<i>Prerequisite</i>	: <i>User</i> harus melakukan <i>login</i> untuk dapat melihat lokasi nya.
<i>Task description</i>	: <i>User</i> dapat melihat lokasi keberadaannya, merekam perjalanannya, merelasikan foto dengan lokasi, merelasikan video dengan lokasi, merelasikan voice record dengan lokasi, merelasikan note dengan lokasi, share dan impor data perjalanan.

### 3.8.2 Rancangan Aplikasi Track My Trip

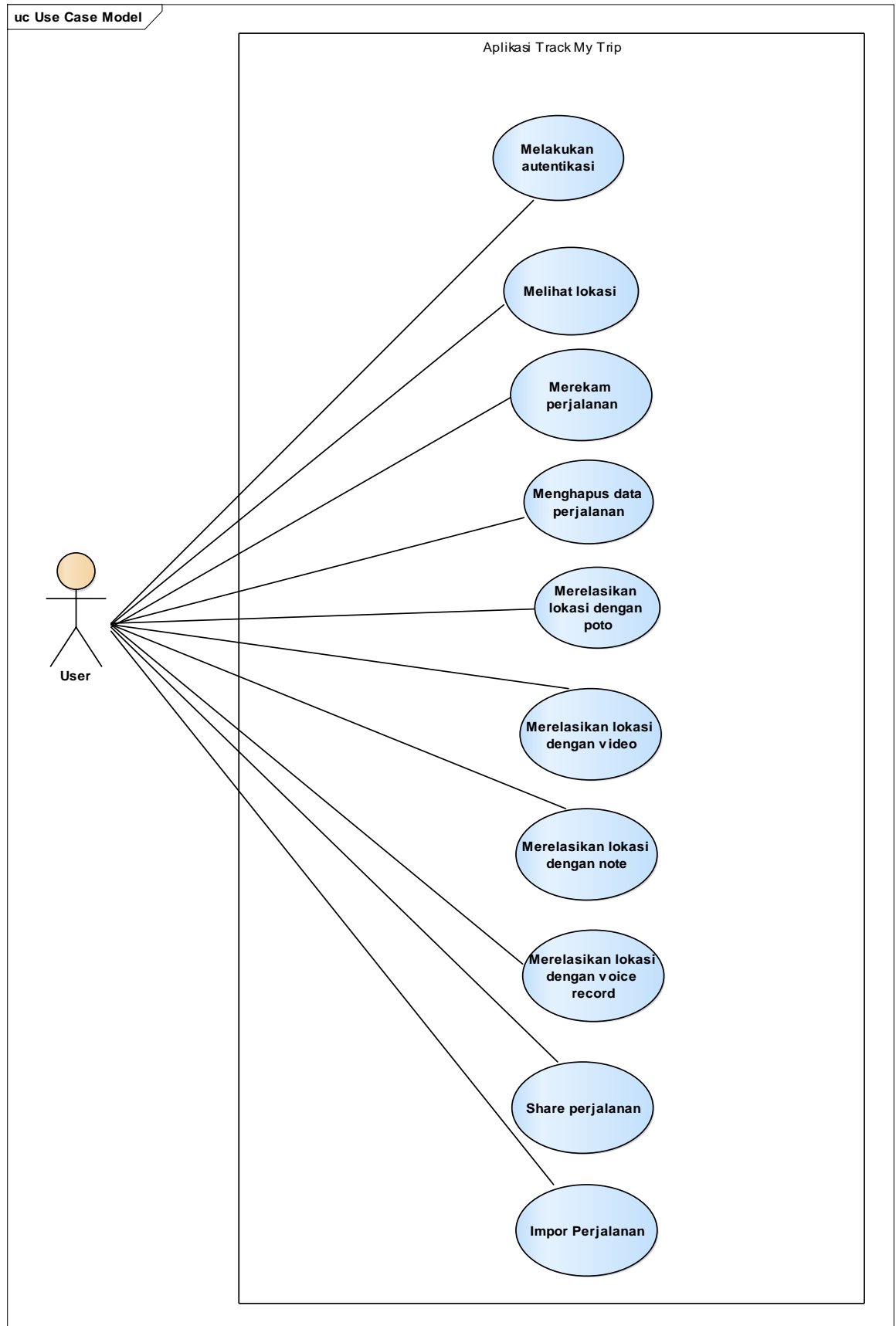
Pada sub bab ini menjelaskan diagram untuk digunakan sebagai rancangan dari aplikasi yang akan dikembangkan yaitu *use case diagram, conceptual data model, physical data model, class diagram, dan sequence diagram*.

#### 3.8.2.1 Use Case Diagram

Pada bagian ini menjelaskan hal apa saja atau kegiatan yang dapat dilakukan oleh aktor pada aplikasi yang ditunjukkan pada *use case diagram*. Dalam *use case diagram* ini terdapat satu aktor yaitu *user* atau pengguna. *Use case diagram* dikelompokkan menjadi satu *boundaries*, yaitu *user* sebagai aktor utama dapat melakukan 10 aktivitas. Aktivitas yang dapat dilakukan antara lain melakukan

autentikasi yaitu *user* register dan login, dapat melihat lokasi atau keberadaannya, *user* dapat merekam perjalanannya, menghapus data perjalanan, *user* dapat merelasikan foto dengan titik koordinatnya, *user* dapat merelasikan video dengan titik koordinatnya, *user* dapat merelasikan voice record dengan titik koordinatnya, *user* dapat merelasikan note dengan titik koordinatnya, *user* dapat membagikan perjalanan yang telah direkam, dan *user* dapat mengimpor data perjalanannya. Pada Gambar 3.28 berikut ini menunjukkan *use case diagram* dari aplikasi.





Gambar 3.28. Use Case Diagram

### 3.8.2.2 Use Case Scenario

Pada sub bab ini dijelaskan *use case scenario* atau use case detail dari seluruh *use case diagram*.

#### 3.8.2.2.1 Use Case Melakukan Autentikasi

Dalam Tabel 3.4 ini terdapat alur kinerja atau langkah untuk *use case* dari melakukan autentikasi.

**Tabel 3.4. Use Case Scenario Melakukan Autentikasi**

Use Case ID	UC_ATT_001	
Use Case Name	Melakukan Autentikasi	
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk melakukan autentikasi	
Primary Actor	<i>User</i>	
Secondary Actor	-	
Pre-condition	<i>User</i> sudah mengakses aplikasi	
Post-condition	<i>User</i> sudah berhasil mendaftarkan akun nya dan login ke dalam aplikasi	
Included Use Case	-	
	<b>Register</b>	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi

	3. <i>User</i> memilih menu register	
		4. Sistem menampilkan form register
	5. <i>User</i> mengisi form register dengan memasukkan nama lengkap, alamat email, nomor hp, password, konfirmasi password	
		6. Sistem menyimpan data yang dimasukkan <i>user</i>
	<b>Login</b>	
	1. <i>User</i> sudah mengakses halaman utama aplikasi track my trip	
		2. Sistem menampilkan halaman utama aplikasi
	3. <i>User</i> memilih menu login	
		4. Sistem menampilkan form
	5. <i>User</i> mengisi form login dengan memasukkan <i>username password</i>	

		6 Sistem menyimpan data yang dimasukkan <i>user</i>
Alternative Flow of Events	5a. Data yang diinput oleh pengguna tidak sesuai dengan ketentuan maka data tidak akan disimpan dan tetap di halaman awal	

### 3.8.2.2.2 Use Case Scenario Menunjukkan Lokasi Pengguna

Dalam Tabel 3.5 ini terdapat alur kinerja atau langkah untuk *use case* dari menunjukkan lokasi pengguna.

**Tabel 3.5. Use Case Scenario Menunjukkan Lokasi Pengguna**

Use Case ID	UC_ATT_003	
Use Case Name	Menunjukkan lokasi pengguna	
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk melihat lokasi keberadaannya berada	
Primary Actor	<i>User</i>	
Secondary Actor	-	
Pre-condition	<i>User</i> sudah mengakses aplikasi	
Post-condition	<i>User</i> berhasil melihat lokasi keberadaannya	
Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi

	3. <i>User</i> mengaktifkan GPS	
		4. Sistem menampilkan peta
	5. <i>User</i> menekan tombol fokus	
		6. Sistem menampilkan lokasi pengguna
Alternative Flow of Events		

### 3.8.2.2.3 Use Case Scenario Merekam Perjalanan

Dalam Tabel 3.6 ini terdapat alur kinerja atau langkah untuk *use case* dari merekam perjalanan.

**Tabel 3.6. Use Case Scenario Merekam Perjalanan**

Use Case ID	UC_ATT_004
Use Case Name	Merekam perjalanan
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk merekam <i>history</i> perjalanan nya
Primary Actor	<i>User</i>
Secondary Actor	-
Pre-condition	<i>User</i> sudah mengakses aplikasi dan masuk ke halaman utama
Post-condition	<i>User</i> berhasil merekam perjalanannya
Included Use Case	-

Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi
	3. <i>User</i> memilih tombol ‘track my journey’	
		4. Sistem mulai merekam perjalanan
	5. <i>User</i> menekan tombol ‘end’	
		6. Sistem akan berhenti melakukan perekaman perjalanan
Alternative Flow of Events		

#### 3.8.2.2.4 Use Case Scenario Menghapus Data Perjalanan

Dalam Tabel 3.7 ini terdapat alur kinerja atau langkah untuk *use case* dari merekam perjalanan.

Tabel 3.7. Use Case Scenario Menghapus Data Perjalanan

Use Case ID	UC_ATT_005	
Use Case Name	Menghapus data perjalanan	
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk menghapus data perjalanannya	
Primary Actor	<i>User</i>	
Secondary Actor	-	
Pre-condition	<i>User</i> sudah mengakses aplikasi	
Post-condition	<i>User</i> berhasil menghapus data perjalanan nya	
Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		1. Sistem menampilkan halaman utama aplikasi
	2. <i>User</i> melihat data perjalanan nya	
		3. Sistem menampilkan data perjalanannya
	4. <i>User</i> menghapus data perjalanannya	
		5. Sistem berhasil menghapus data perjalanan

Alternative Flow of Events	
----------------------------	--

### 3.8.2.2.5 Use Case Scenario Merelasikan Lokasi dengan Foto

Dalam Tabel 3.8 ini terdapat alur kinerja atau langkah untuk *use case* dari merelasikan foto dengan lokasi titik pengambilannya.

**Tabel 3.8. Use Case Scenario Merelasikan Lokasi dengan Foto**

Use Case ID	UC_ATT_006	
Use Case Name	Merelasikan lokasi dengan foto	
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk merelasikan foto yang diambilnya dengan lokasi titik tempat pengguna mengambilnya	
Primary Actor	<i>User</i>	
Secondary Actor	-	
Pre-condition	<i>User</i> sudah mengakses aplikasi dan masuk ke halaman utama	
Post-condition	<i>User</i> berhasil merelasikan foto	
Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi



	3. <i>User</i> memilih menu kamera	
		4. Sistem mengaktifkan kamera
	5. <i>User</i> mengambil foto	
		6. Sistem menyimpan foto yang telah diambil
Alternative Flow of Events		

#### 3.8.2.2.6 Use Case Scenario Merelasikan Lokasi dengan Video

Dalam Tabel 3.9 ini terdapat alur kinerja atau langkah untuk *use case* dari merelasikan video dengan lokasi titik pengambilannya.

**Tabel 3.9. Use Case Scenario Merelasikan Lokasi dengan Video**

Use Case ID	UC_ATT_007
Use Case Name	Merelasikan lokasi dengan video
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk merelasikan video yang diambilnya dengan lokasi titik tempat pengguna mengambilnya
Primary Actor	<i>User</i>
Secondary Actor	-
Pre-condition	<i>User</i> sudah mengakses aplikasi dan masuk ke halaman utama
Post-condition	<i>User</i> berhasil merelasikan video

Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi
	3. <i>User</i> memilih menu kamera	
		4. Sistem mengaktifkan kamera
	5. <i>User</i> mengambil video	
		6. Sistem menyimpan video yang telah diambil
Alternative Flow of Events		

### 3.8.2.2.7 Use Case Scenario Merelasikan Lokasi dengan Video

Dalam Tabel 3.10 ini terdapat alur kinerja atau langkah untuk *use case* dari merelasikan note dengan lokasi titik pengambilannya.

**Tabel 3.10. Use Case Scenario Merelasikan Lokasi dengan Note**

Use Case ID	UC_ATT_008
Use Case Name	Merelasikan lokasi dengan note

Brief Description	Use case ini menjelaskan proses <i>user</i> untuk merelasikan note atau catatan yang ditulisnya dengan lokasi titik tempat pengguna mengambilnya	
Primary Actor	<i>User</i>	
Secondary Actor	-	
Pre-condition	<i>User</i> sudah mengakses aplikasi dan masuk ke halaman utama	
Post-condition	<i>User</i> berhasil merelasikan note	
Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi
	3. <i>User</i> memilih menu note	
		3 Sistem menampilkan note
	4 <i>User</i> menulis note	
		5 Sistem menyimpan note yang telah ditulis
Alternative Flow of Events		

### 3.8.2.2.8 Use Case Merelasikan Lokasi dengan Voice Record

Dalam Tabel 3.11 ini terdapat alur kinerja atau langkah untuk *use case* dari merelasikan voice record dengan lokasi titik pengambilannya.

**Tabel 3.11. Use Case Scenario Merelasikan Lokasi dengan Voice Record**

Use Case ID	UC_ATT_009	
Use Case Name	Merelasikan lokasi dengan note	
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk merelasikan suara pengguna dengan lokasi titik tempat pengguna mengambilnya	
Primary Actor	<i>User</i>	
Secondary Actor	-	
Pre-condition	<i>User</i> sudah mengakses aplikasi dan masuk ke halaman utama	
Post-condition	<i>User</i> berhasil merelasikan voice record	
Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi
	3. <i>User</i> memilih menu voice	
		4. Sistem menampilkan menu voice record

	5. <i>User</i> merekam suara	
		6.Sistem menyimpan hasil rekaman suara
Alternative Flow of Events		

### 3.8.2.2.9 Use Case Scenario Share perjalanan

Dalam Tabel 3.12 ini terdapat alur kinerja atau langkah untuk *use case* dari share perjalanan.

**Tabel 3.12. Use Case Scenario Share Perjalanan**

Use Case ID	UC_ATT_010	
Use Case Name	Share data perjalanan	
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk membagikan data perjalanan	
Primary Actor	<i>User</i>	
Secondary Actor	-	
Pre-condition	<i>User</i> sudah mengakses aplikasi dan masuk ke halaman utama	
Post-condition	<i>User</i> berhasil membagikan perjalanan	
Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman	

	utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi
	3. <i>User</i> melihat data perjalanan	
		4. Sistem menampilkan data perjalanan
	5. <i>User</i> memilih layanan Share	
		6. Sistem membagikan data perjalanan
Alternative Flow of Events		

### 3.8.2.2.10 Use Case Scenario Impor Data Perjalanan

Dalam Tabel 3.13 ini terdapat alur kinerja atau langkah untuk *use case* dari impor perjalanan.

**Tabel 3.13. Use Case Scenario Impor Data Perjalanan**

Use Case ID	UC_ATT_011
Use Case Name	Impor data perjalanan
Brief Description	Use case ini menjelaskan proses <i>user</i> untuk mengimpor data perjalanan
Primary Actor	<i>User</i>
Secondary Actor	-

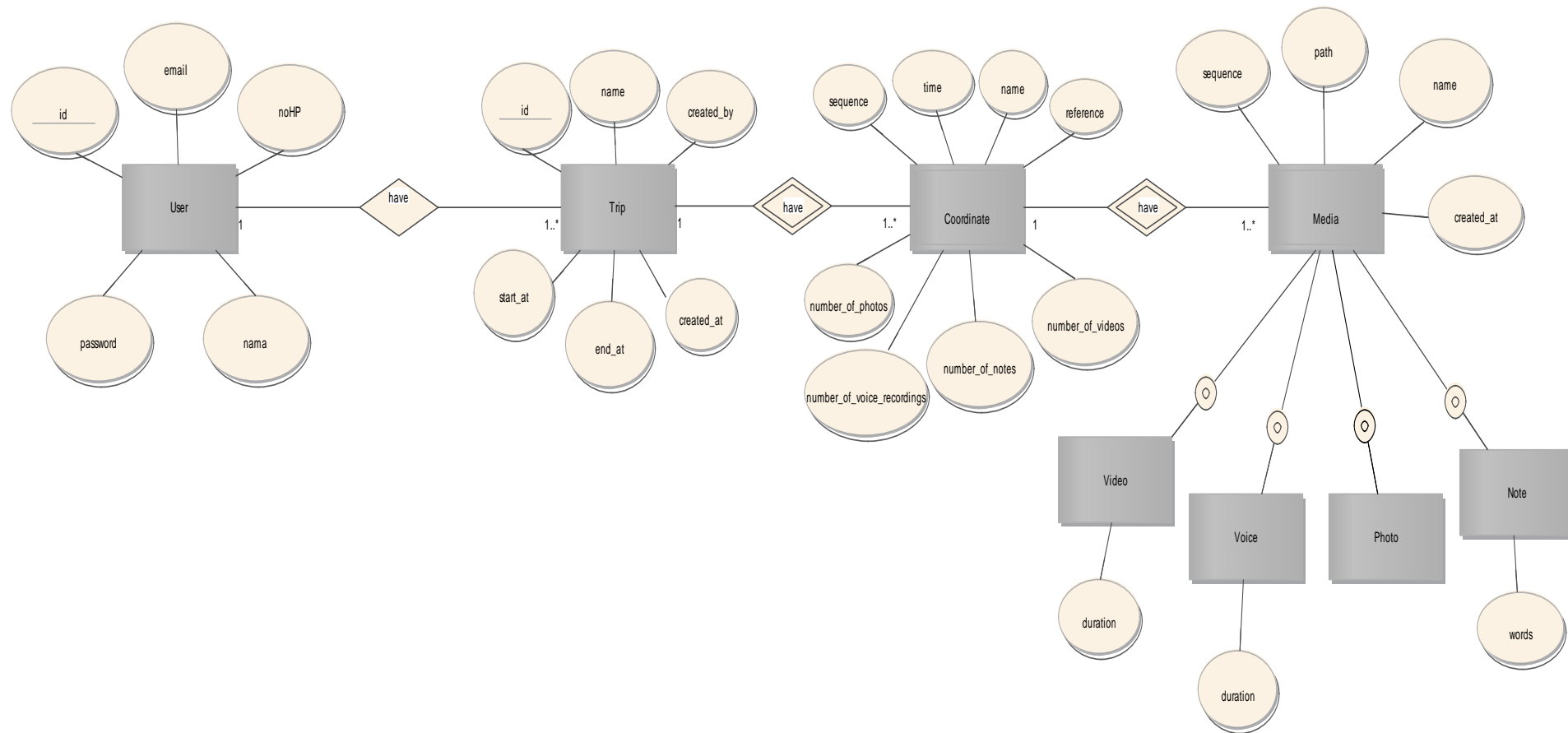
Pre-condition	<i>User</i> sudah mengakses aplikasi dan masuk ke halaman utama	
Post-condition	<i>User</i> berhasil melakukan mengimpor data perjalanan	
Included Use Case	-	
Basic Flow of Events	<i>User Action</i>	<i>System Response</i>
	1. <i>User</i> sudah mengakses halaman utama aplikasi <i>track my trip</i>	
		2. Sistem menampilkan halaman utama aplikasi
	3. <i>User</i> melihat data perjalanan	
		4. Sistem menampilkan data perjalanan
	5. <i>User</i> memilih layanan impor	
		6. Sistem meminta file yang akan diimpor

### 3.8.3 Rancangan Kebutuhan Data

Pada sub bab ini dijelaskan mengenai kebutuhan data dari aplikasi yang digunakan untuk proses pengembangan aplikasi yaitu dalam bentuk ERD. Pada desain ERD ini terdapat 8 *entitas*, yaitu *user*, *trip*, *coordinate*, *media*, *photo*, *video*, *voice*, dan *note*. Trip memiliki atribut id sebagai *primary key*, *name*, *created\_by*, *created\_at*, *start\_at*, *end\_at*. Pada Gambar 3.29 berikut ini menunjukkan ERD dari aplikasi.



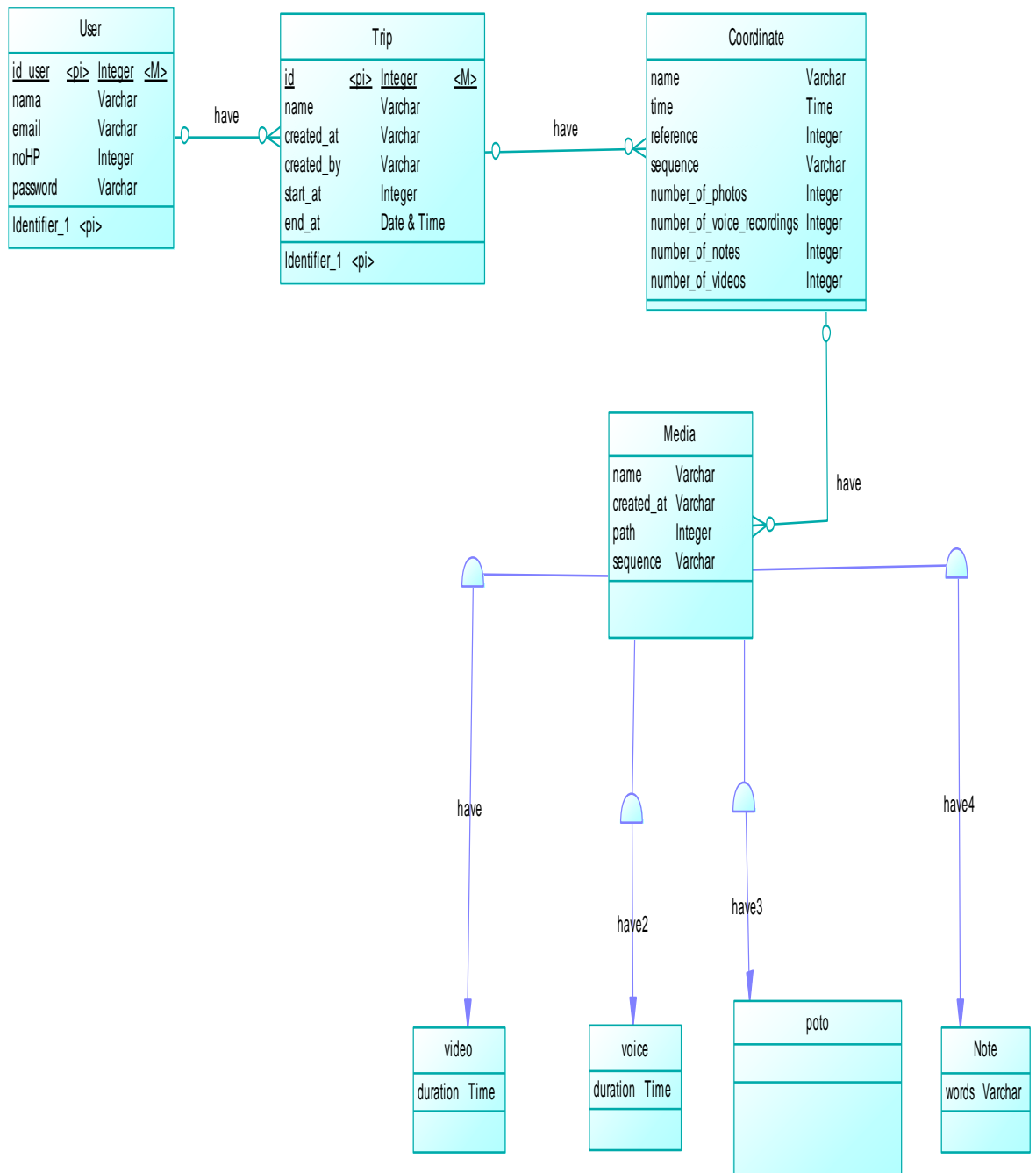
erd Entity Relationship Diagram



Gambar 3.29. Entity Relationship Diagram

### 3.8.3.1 Conceptual Data Model

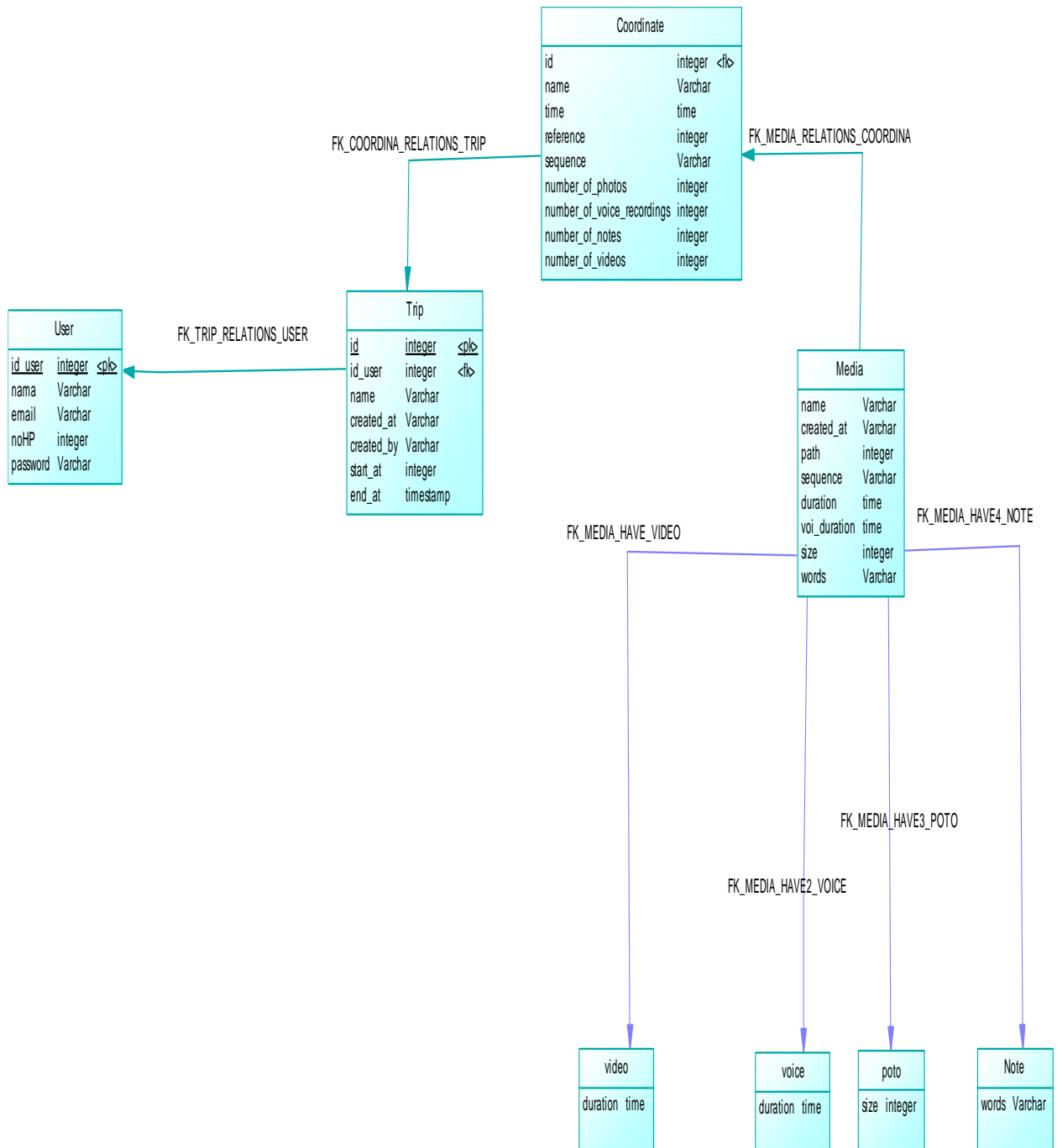
Pada sub bab ini digambarkan *conceptual data model* aplikasi Track My Trip. Pada *conceptual data model* ini terdiri dari objek-objek dasar berupa *entity* serta hubungan antara entitas-entitas tersebut. Pada Gambar 3.30 berikut ini menunjukkan CDM aplikasi.



Gambar 3.30. Conceptual Data Model

### 3.8.3.2 Physical Data Model

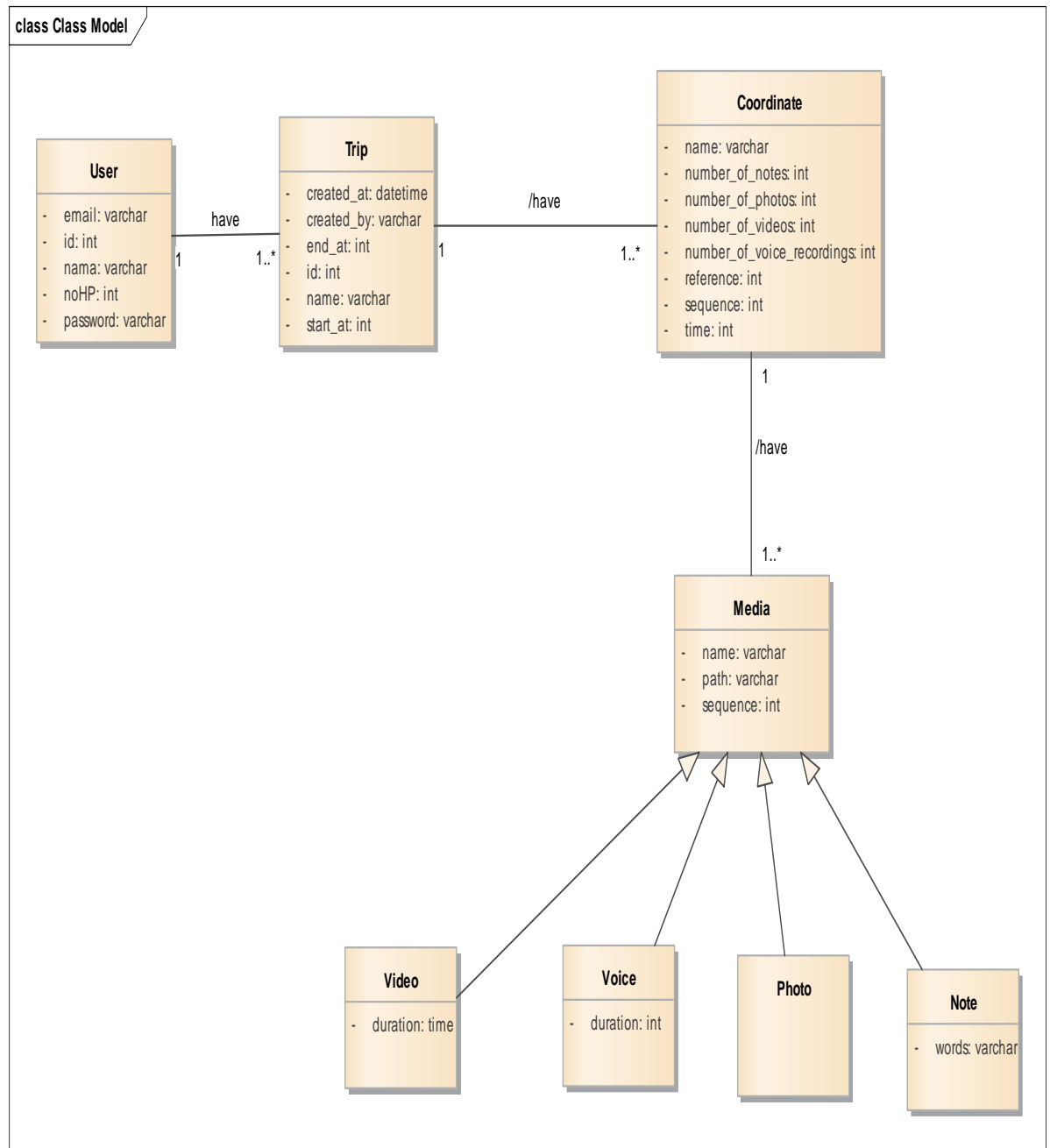
Pada sub bab ini digambarkan *physical data model* aplikasi *track my trip*. Pada *conceptual data model* ini terdiri dari objek-objek dasar berupa data serta hubungan antara data-data tersebut. Pada Gambar 3.31 berikut ini menunjukkan PDM aplikasi.



Gambar 3.31. Physical Data Model

### 3.8.3.3 Class Diagram

Pada sub bab ini digambarkan *class diagram* aplikasi *track my trip*. Pada Gambar 3.32 berikut ini menunjukkan class diagram aplikasi.



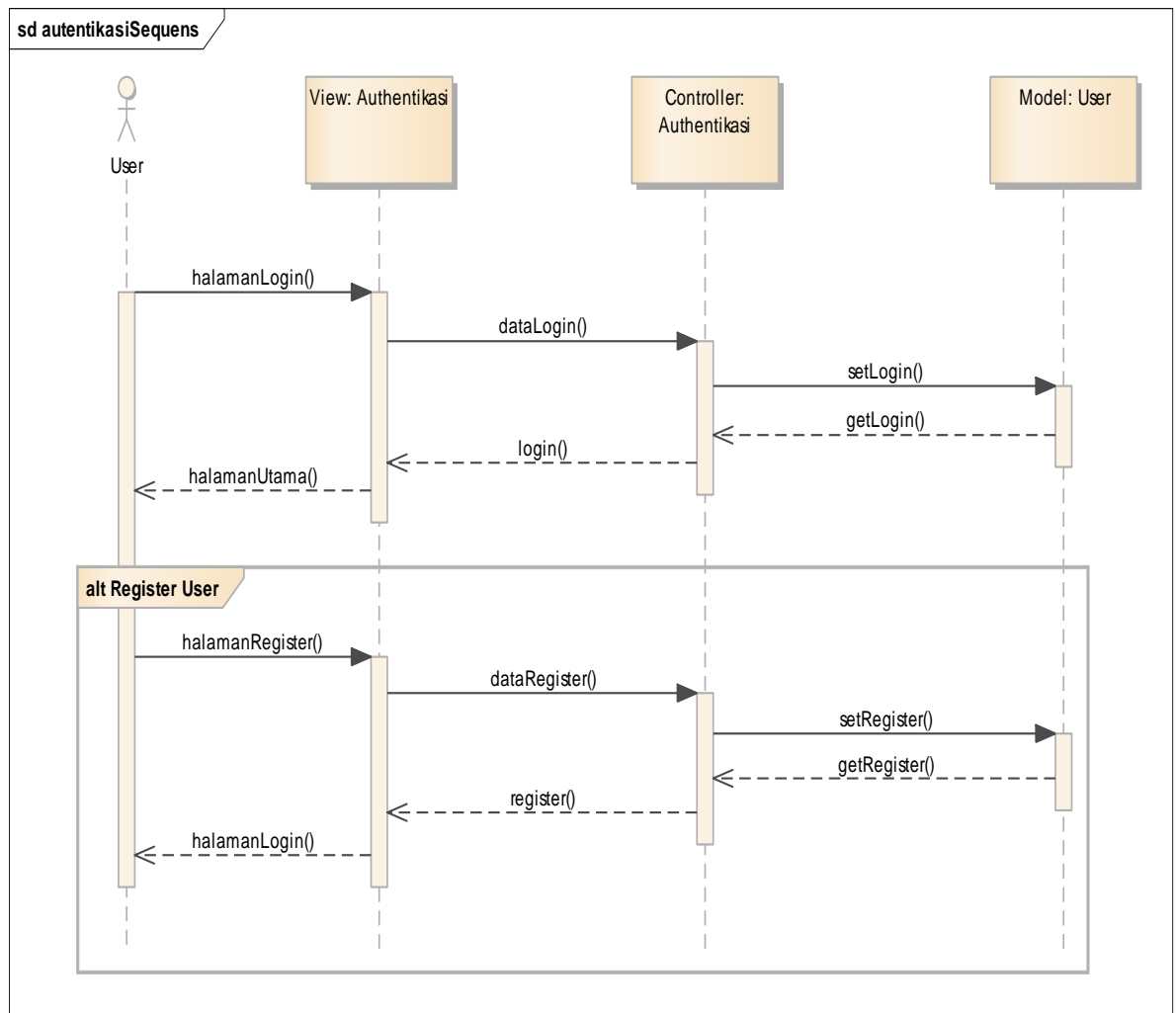
Gambar 3.32. Class Diagram

### 3.8.3.4 Sequence Diagram

Pada sub bab ini dijelaskan mengenai urutan proses yang terjadi pada aplikasi *track my trip* dan menampilkan rangkaian pesan yang disampaikan suatu objek dapat berinteraksi dengan objek lainnya.

#### 3.8.3.4.1 Melakukan Autentikasi

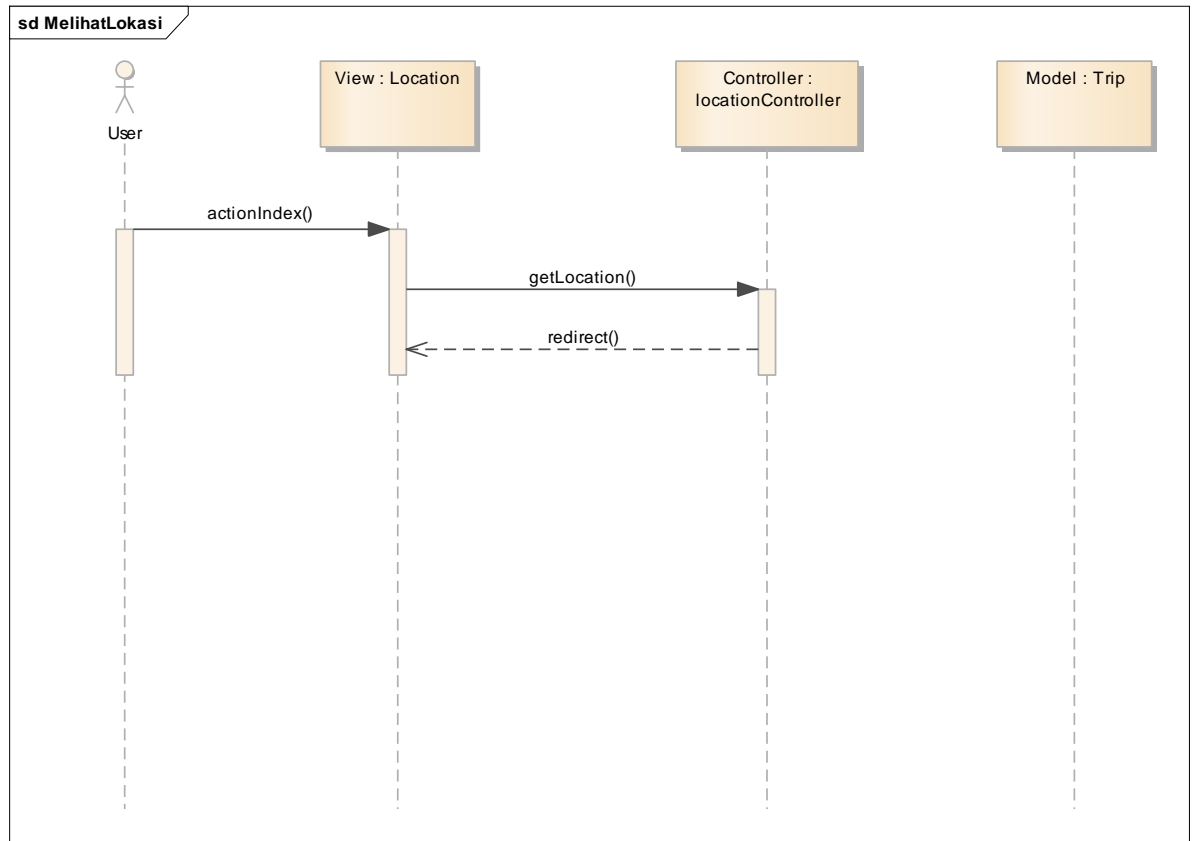
Pada *sequence diagram* ini dijelaskan alur untuk melakukan autentikasi. *User* yang ingin merekam perjalanan nya terlebih dahulu masuk mendaftarkan akunnya ke dalam aplikasi. Ketika pengguna telah berhasil mendaftarkan akunnya, maka pengguna akan melakukan login ke dalam aplikasi. Urutan diagram untuk melakukan autentikasi dapat dilihat pada Gambar 3.33.



Gambar 3.33. Sequence Diagram Melakukan Autentikasi

### 3.8.3.4.2 Menunjukkan Lokasi Pengguna

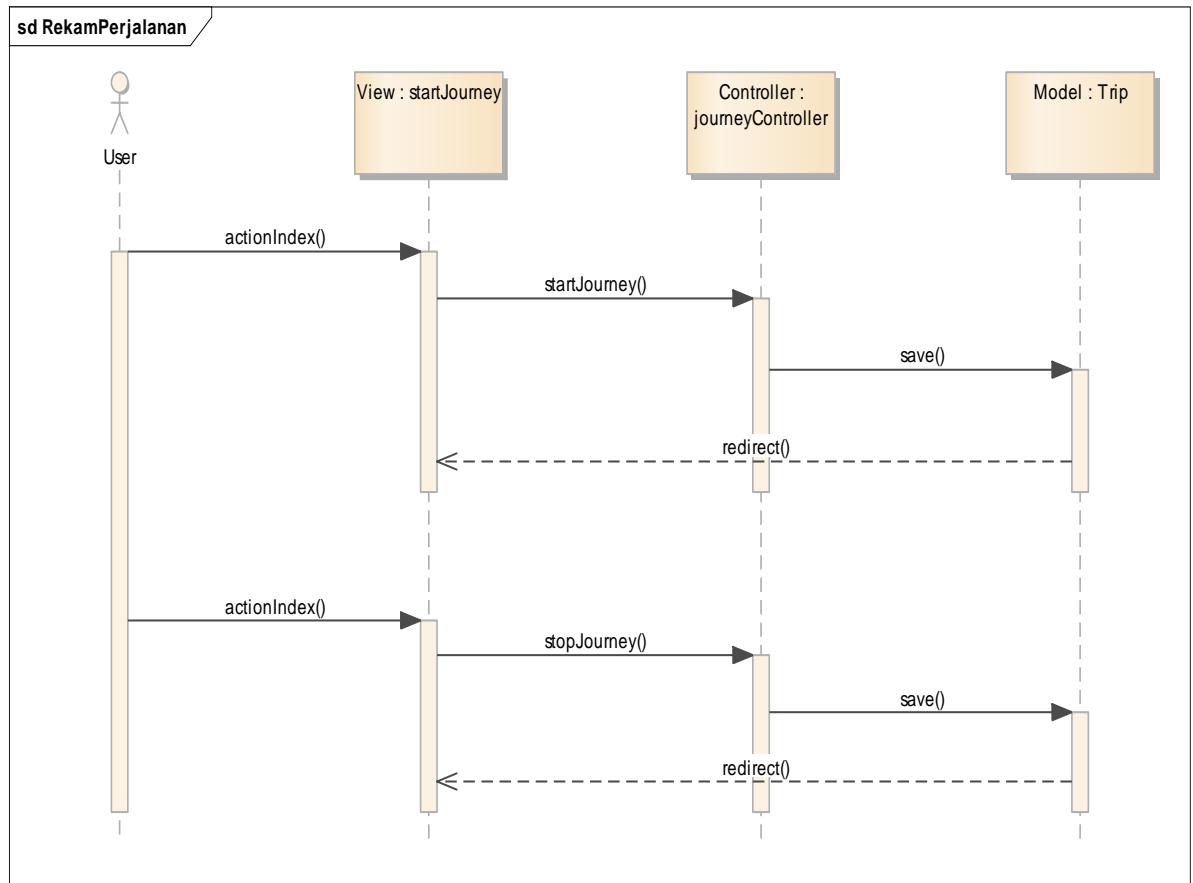
Pada *sequence diagram* ini dijelaskan alur untuk melihat lokasi yang dilakukan oleh *user*. *User* yang ingin melihat lokasinya dapat melihat *sequence diagram* di bawah ini. *User* yang ingin melihat lokasinya dapat masuk ke halaman utama terlebih dahulu. Pada Gambar 3.34 berikut ini menunjukkan *sequence diagram* menunjukkan lokasi pengguna.



Gambar 3.34. Sequence Diagram Lokasi Pengguna

### 3.8.3.4.3 Rekam Perjalanan

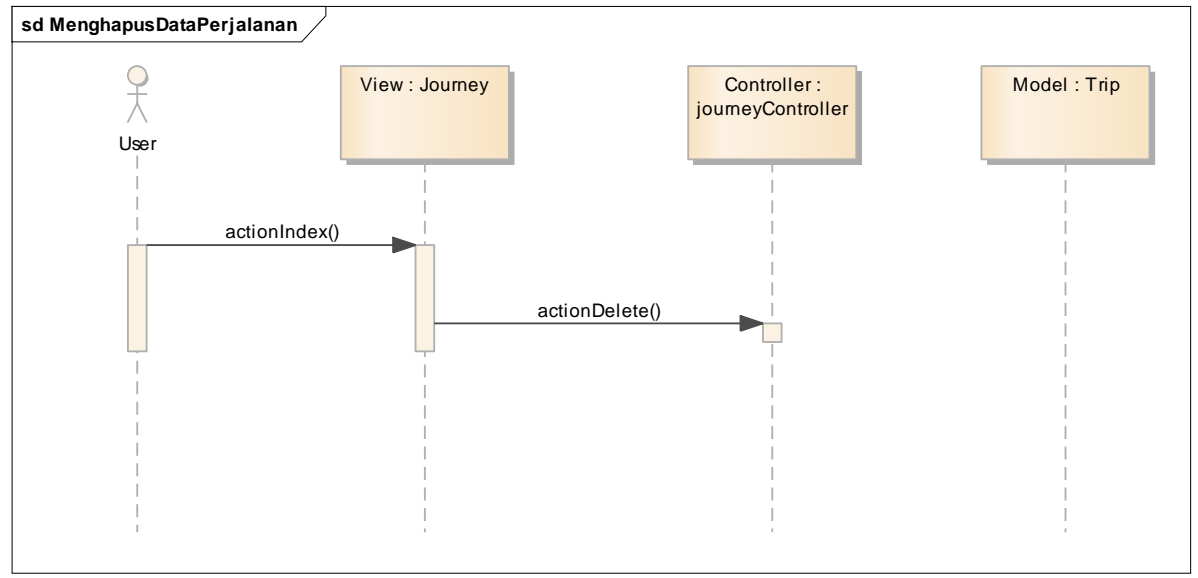
Pada *sequence diagram* ini dijelaskan alur untuk merekam perjalanan. *User* yang ingin merekam perjalanan nya terlebih dahulu masuk ke halaman utama. Kemudian menekan tombol “*start*” untuk memulai perekaman perjalanan dan tombol “*stop*” untuk memberhentikan perekaman perjalanan. Urutan diagram untuk merekam perjalanan dapat dilihat pada Gambar 3.35.



Gambar 3.35. Sequence Diagram Rekam Perjalanan

#### 3.8.3.4.4 Menghapus Data Perjalanan

Pada *sequence diagram* ini dijelaskan alur untuk menghapus perjalanan yang telah direkam. *User* dapat menghapus data perjalanan dengan terlebih dahulu melihat daftar perjalanannya, setelah *user* menekan tombol hapus, maka data perjalanan yang dipilih akan terhapus. Urutan diagram untuk menghapus data perjalanan dapat dilihat pada Gambar 3.36.

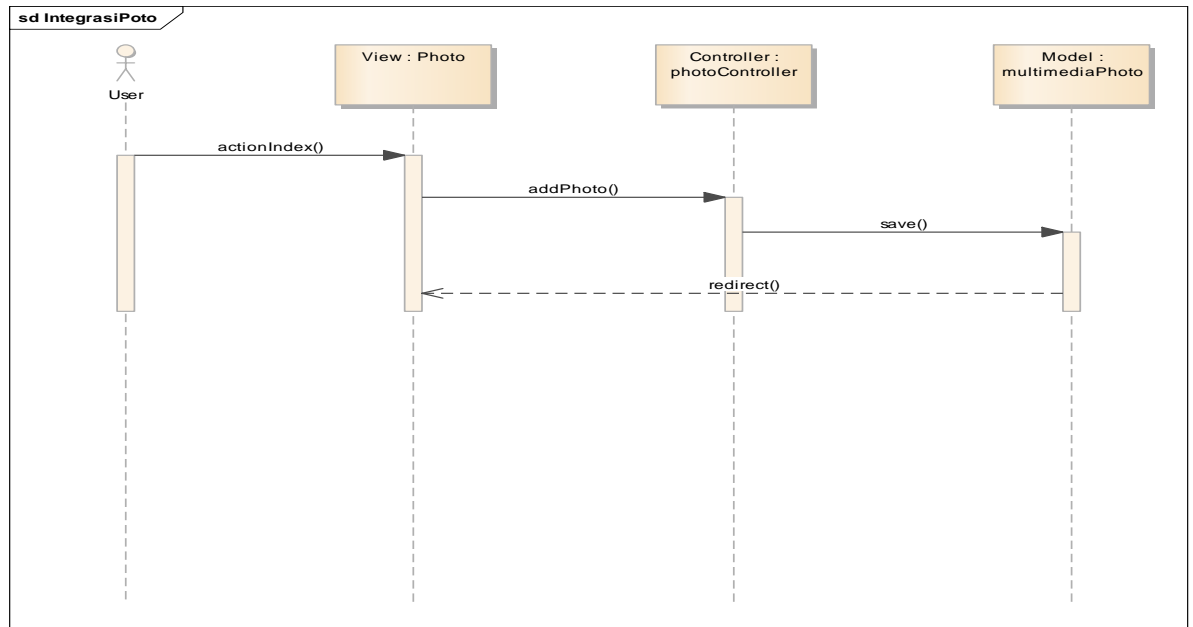


Gambar 3.36. Sequence Diagram Menghapus Data Perjalanan

#### 3.8.3.4.5 Merelasikan Lokasi dengan Foto

Pada *sequence diagram* ini dijelaskan alur untuk merelasikan foto dengan lokasi pengguna berada. *User* yang ingin merelasikan foto dengan titik koordinat keberadaan pengguna terlebih dahulu memilih menu foto, kemudian mengambil foto. Setelah foto diambil, pengguna akan bisa langsung menyimpan foto atau tidak menyimpan foto tersebut. Pada Gambar 3.37 berikut ini menunjukkan *sequence diagram* menunjukkan merelasikan lokasi dengan foto.

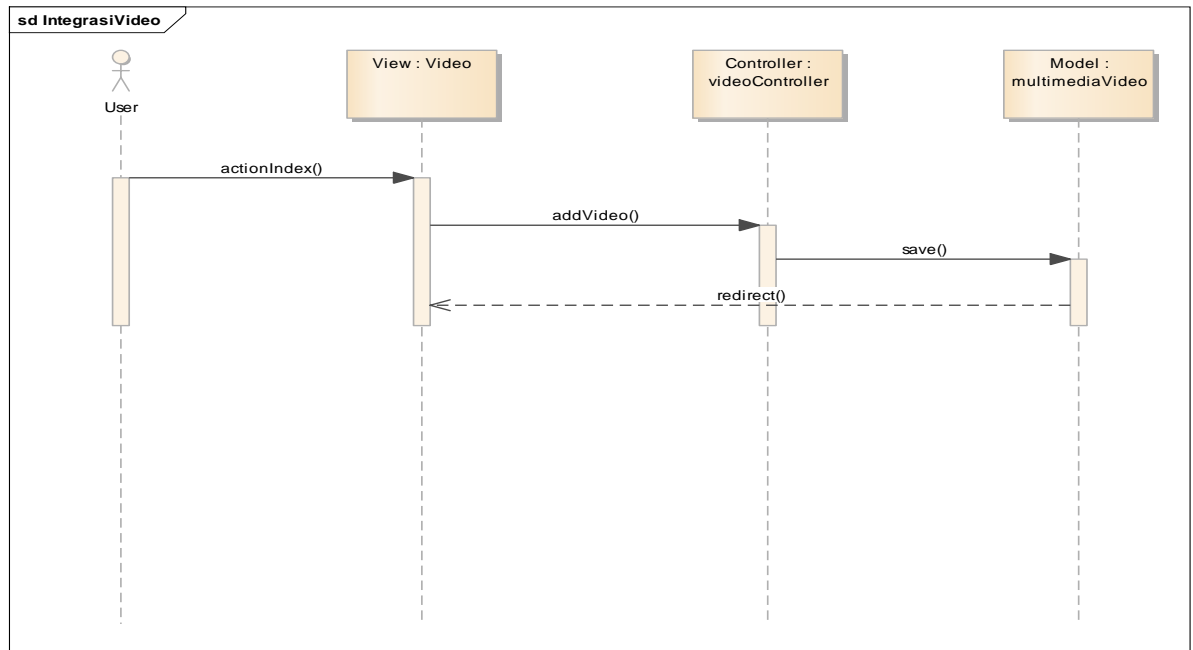




Gambar 3.37. Sequence Diagram Merelasikan Lokasi dengan Foto

#### 3.8.3.4.6 Merelasikan Lokasi dengan Video

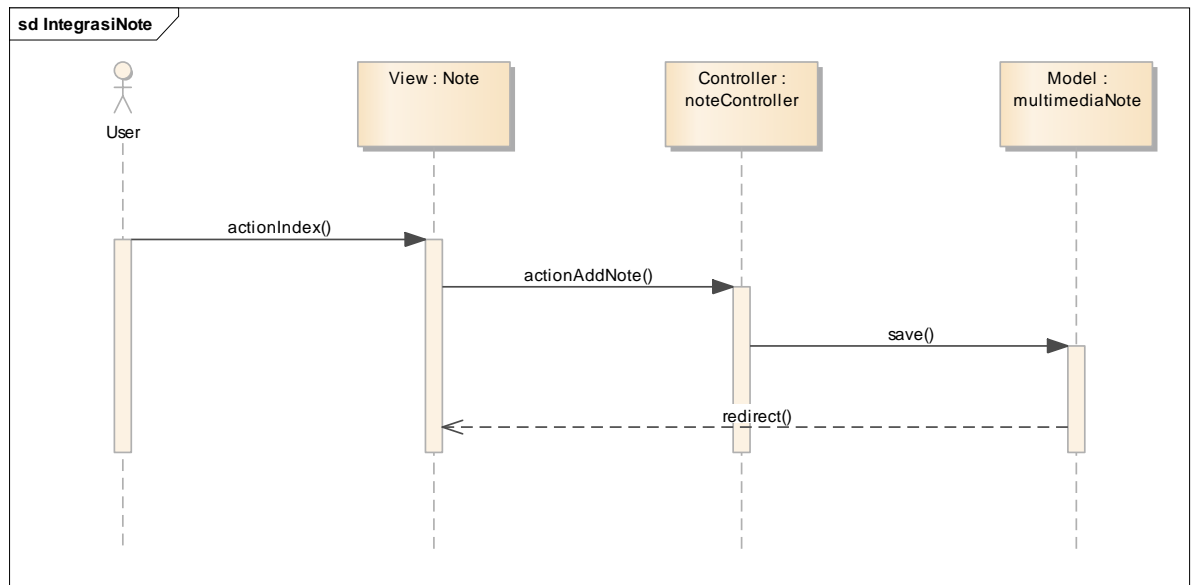
Pada *sequence diagram* ini dijelaskan alur untuk merelasikan video dengan lokasi pengguna berada. *User* yang ingin merelasikan video dengan titik koordinat keberadaan pengguna terlebih dahulu memilih menu video, kemudian merekam video dan menyimpan hasil rekaman video yang telah direkam. Pada Gambar 3.38 berikut ini menunjukkan *sequence diagram* menunjukkan merelasikan lokasi dengan video.



**Gambar 3.38. Sequence Diagram Merelasikan Lokasi dengan Video**

#### 3.8.3.4.7 Merelasikan Lokasi dengan Note

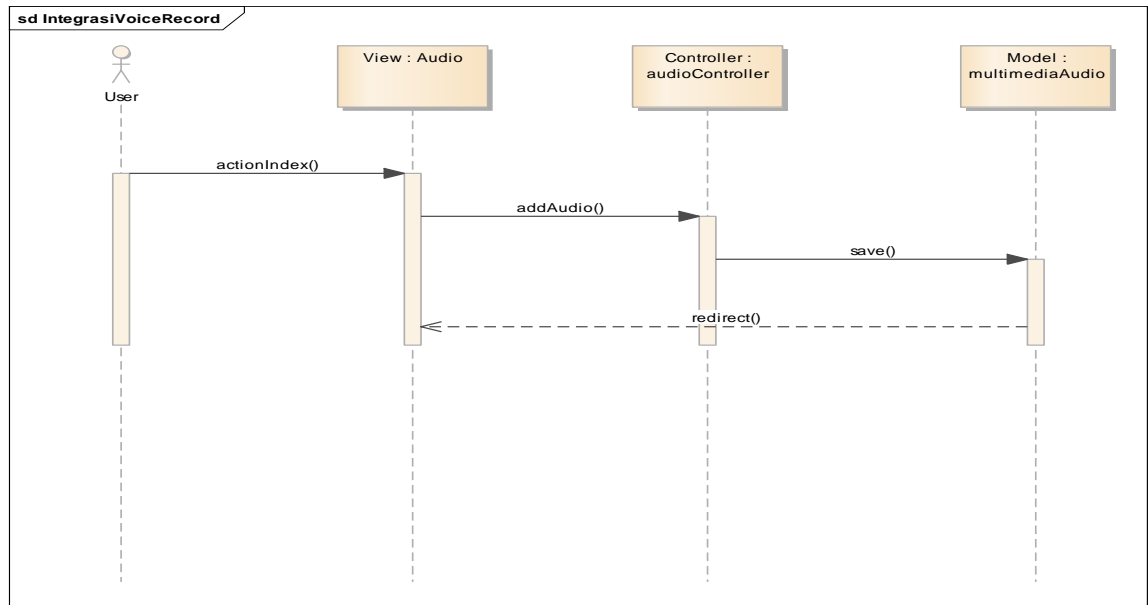
Pada *sequence diagram* ini dijelaskan alur untuk merelasikan note dengan lokasi pengguna berada. *User* yang ingin merelasikan note dengan titik koordinat keberadaan pengguna terlebih dahulu memilih menu note, kemudian menulis note. Setelah note ditulis, pengguna akan bisa langsung menyimpan note atau tidak menyimpan note tersebut. Urutan diagram untuk merelasikan note dapat dilihat pada Gambar 3.39 ini.



**Gambar 3.39. Sequence Diagram Merelasikan Lokasi dengan Note**

#### 3.8.3.4.8 Merelasikan Lokasi dengan Voice Record

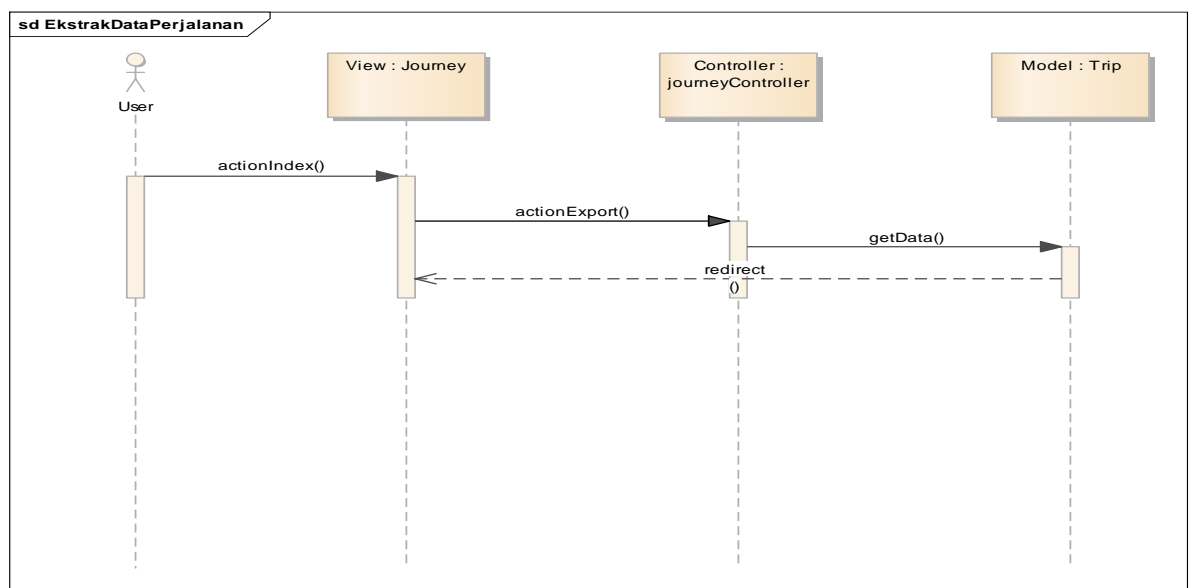
Pada *sequence diagram* ini dijelaskan alur untuk merelasikan voice record dengan lokasi pengguna berada. *User* yang ingin merelasikan suara dengan titik koordinat keberadaan pengguna terlebih dahulu memilih menu voice record, kemudian merekam suara. Setelah voice direkam, pengguna akan bisa langsung menyimpan voice atau tidak menyimpan voice tersebut. Urutan diagram untuk merelasikan voice record dapat dilihat pada Gambar 3.40.



Gambar 3.40. Sequence Diagram Merelasikan Lokasi dengan Voice Record

### 3.8.3.4.9 Share Perjalanan

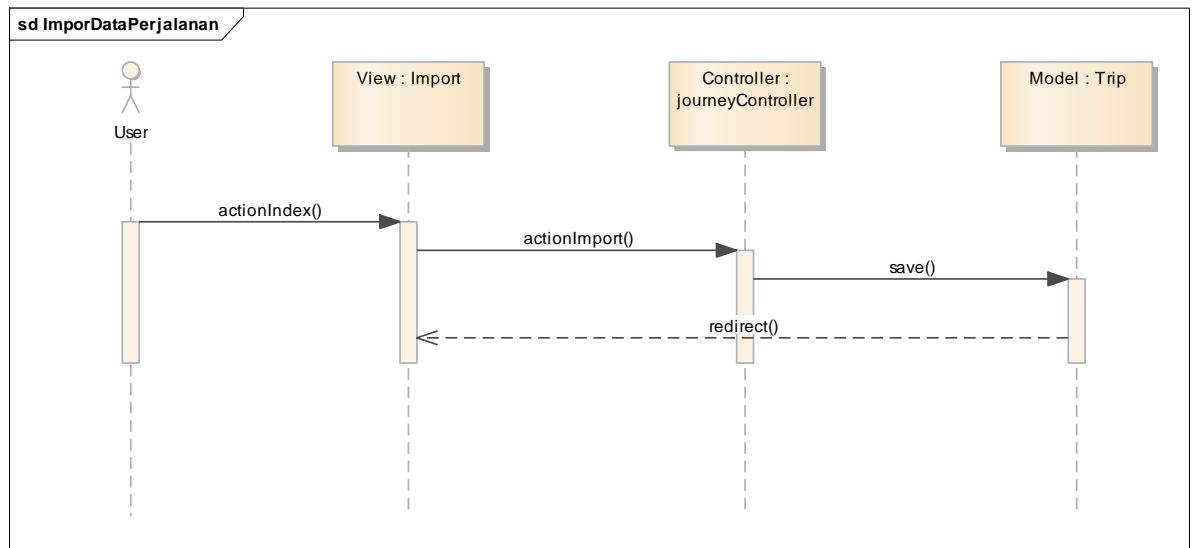
Pada *sequence diagram* ini dijelaskan alur membagikan perjalanan yang telah direkam. *User* dapat membagikan data perjalanan yang telah ada dalam bentuk data yang telah diekstrak yaitu bentuk Zip. Urutan diagram untuk membagikan perjalanan dapat dilihat pada Gambar 3.41.



Gambar 3.41. Sequence Diagram Share perjalanan

### 3.8.3.4.10 Impor Data Perjalanan

Pada *sequence diagram* ini dijelaskan alur mengimpor data perjalanan yang telah direkam. *User* dapat mengimpor data perjalanan yang telah ada dalam bentuk data yang telah diekstrak yaitu bentuk Zip. Urutan diagram untuk melakukan impor data perjalanan dapat dilihat pada Gambar 3.42.



Gambar 3.42. Sequence Diagram Impor Data Perjalanan

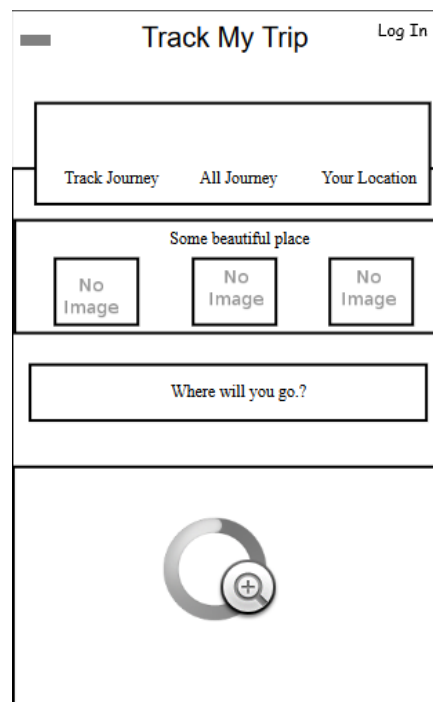
### 3.8.3.5 Desain User Interface Aplikasi

Pada sub bab ini akan menjelaskan rancangan antarmuka aplikasi *track my trip* yakni mulai dari rancangan antarmuka home, perjalanan, merelasikan lokasi dengan foto, video, voice, dan note. Desain aplikasi *track my trip* dikembangkan dengan tipe desain *user interface* Flat, yang saat ini banyak diterapkan dalam berbagai aplikasi yang sudah terkenal seperti bukalapak, go-jek, tokopedia dll. Flat design memiliki ciri khas pada penggunaan ikon dan ilustrasi. Setiap elemen dalam desain berpusat pada ide yang telah ditentukan dan mempertahankan konsep kesederhanaan. Dalam hal warna, flat design dapat disesuaikan dengan berbagai warna, tetapi pada umumnya flat design menggunakan berbagai warna yang berani dan cerah. Konsep dan warna yang digunakan dalam desain *user interface* aplikasi akan di kembangkan dengan kombinasi warna hijau, warna putih dan abu-abu serta icon flat. Warna hijau dipilih sebagai warna utama dalam aplikasi ini karena

cenderung lebih alami dan tidak membosankan untuk dilihat, warna latar belakang yang dipilih adalah warna putih yang merupakan warna dengan komposisi paling banyak digunakan karena warna putih netral dan cocok untuk dikombinasikan dengan warna cerah, flat icon digunakan karena bentuk dan tampilan yang menarik, tidak membosankan, serta mudah untuk memahami fungsi yang diwakilkan di dalamnya, warna abu-abu merupakan warna yang cocok untuk dikombinasikan dengan warna putih dan dapat terlihat menyatu dengan warna lainnya dan membuat desain tampilan terlihat menarik.

#### 3.8.3.5.1 Desain *User Interface* Home

Pada bagian ini ditampilkan desain *user interface home* menggambarkan menu utama aplikasi *track my trip*. Pada halaman utama akan ditampilkan fungsi yang terdapat dalam aplikasi. Pada halaman utama terdapat icon yang menunjukkan fungsi yang terdapat dalam aplikasi meliputi fungsi untuk merekam perjalanan, menentukan lokasi, dan juga untuk melihat data perjalanan yang telah dilalui. Pada gambar berikut ini menunjukkan desain *user interface* home. Pada Gambar 3.43 terdapat desain tampilan home.



Gambar 3.43. Design UI Home

### 3.8.3.5.2 Desain *User Interface* Register

Desain *user interface register* menggambarkan fungsional saat user akan melakukan registrasi akun. Setiap pengguna harus mendaftarkan akun agar dapat menggunakan fungsi yang dalam aplikasi secara keseluruhan, user dapat melakukan registrasi dengan mendaftarkan akun aplikasi, untuk melakukan pendaftaran user memerlukan alamat email sebagai yang berguna sebagai username, nomor *handphone*, dan membuat password untuk akun yang telah didaftarkan. Pada Gambar 3.44 terdapat desain tampilan register.

The image shows a mobile application interface for a registration screen. At the top, there is a header bar with the text 'Track My Trip' and a 'Log Out' button on the right. Below the header, there is a large rectangular button labeled 'Register'. Underneath this button is a form containing five input fields, each with a label above it: 'nama lengkap', 'alamat Email', 'No HP', 'Password', and 'Confirm Password'. At the bottom of the form, there is a 'Register' button.

**Gambar 3.44. Desain UI Register**

### 3.8.3.5.3 Desain *User Interface* Login

Desain *user interface login* menggambarkan fungsional saat user akan masuk ke dalam aplikasi. Setiap pengguna harus memasukkan username yaitu alamat email dan password. Pada Gambar 3.45 terdapat desain tampilan login.

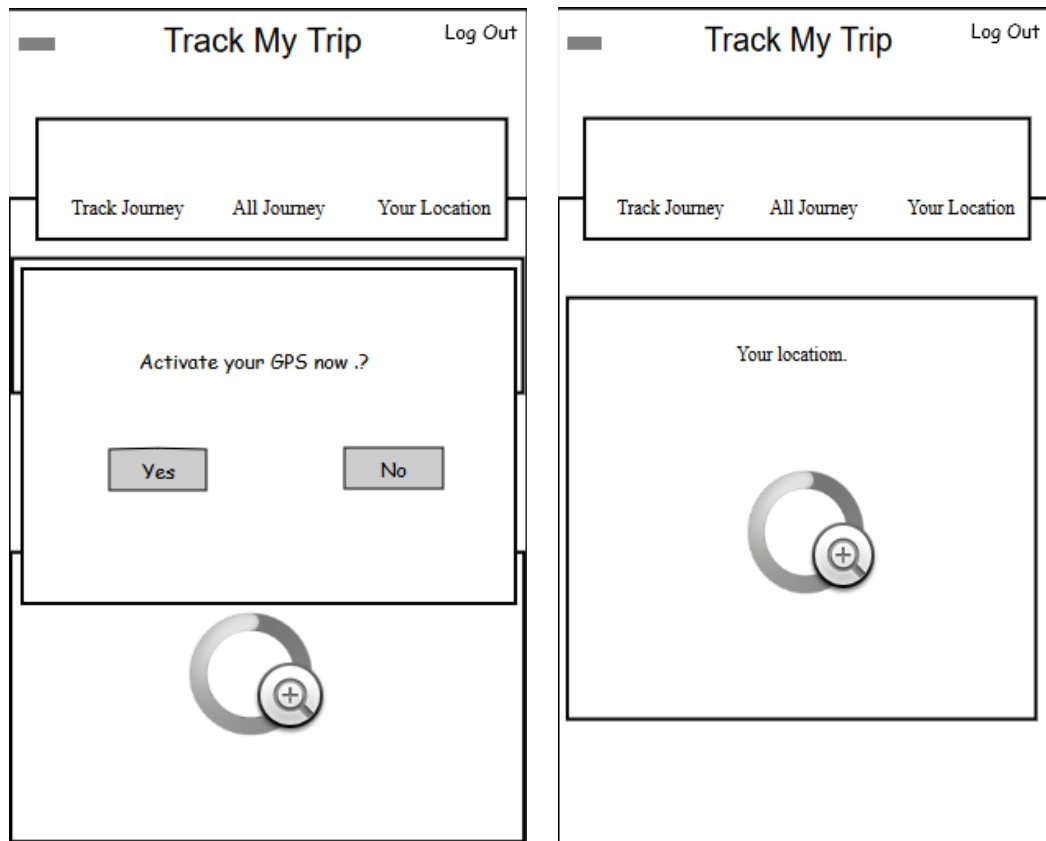
The image shows a mobile application interface for 'Track My Trip'. At the top, there is a header bar with a hamburger menu icon on the left, the title 'Track My Trip' in the center, and a 'Log Out' link on the right. Below the header, there are two main sections. The first section is a registration form with a single button labeled 'Register'. The second section is a login form, which is enclosed in a rounded rectangle. It contains two input fields: the first is labeled 'alamat Email' and the second is labeled 'Password'. Below these fields is a button labeled 'Login'.

**Gambar 3.45. Desain UI Login**

#### **3.8.3.5.4 Desain *User Interface* Lokasi dari Pengguna**

Desain *user interface* lokasi menggambarkan desain untuk menampilkan titik lokasi dimana pengguna berada. Sebelum lokasi ditunjukkan, maka akan muncul *permission* untuk mengaktifkan GPS. Pada Gambar 3.46 berikut ini menunjukkan desain *user interface* menunjukkan lokasi dari pengguna.

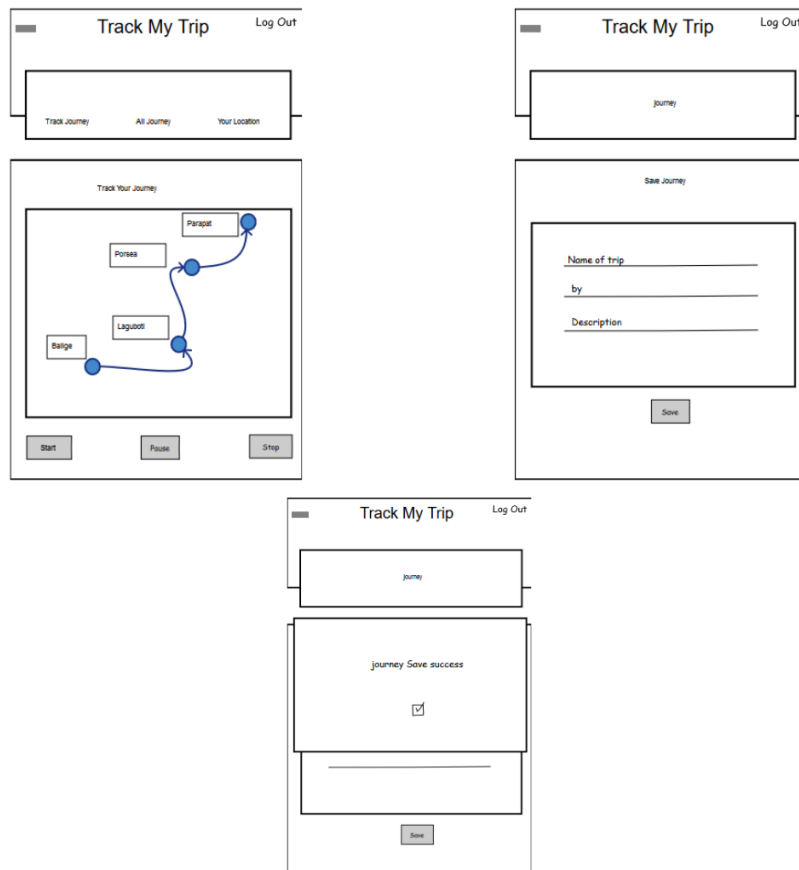




Gambar 3.46. Design UI Lokasi Pengguna

#### 3.8.3.5.5 Desain *User Interface* Rekam Perjalanan

Desain *user interface* perjalanan dan daftar perjalanan dari pengguna, desain ini menggambarkan fungsional untuk melakukan perekaman perjalanan oleh pengguna, design ini menggambarkan fungsi perekaman perjalanan oleh *user* dimana *user* juga dapat melakukan perekaman perjalanan serta menyimpan data multimedia secara langsung seperti gambar, video, audio, dan juga notes yang akan tersimpan otomatis dalam aplikasi. Pada gambar berikut ini menunjukkan desain *user interface* perjalanan. Pada Gambar 3.47 terdapat desain tampilan rekam perjalanan.

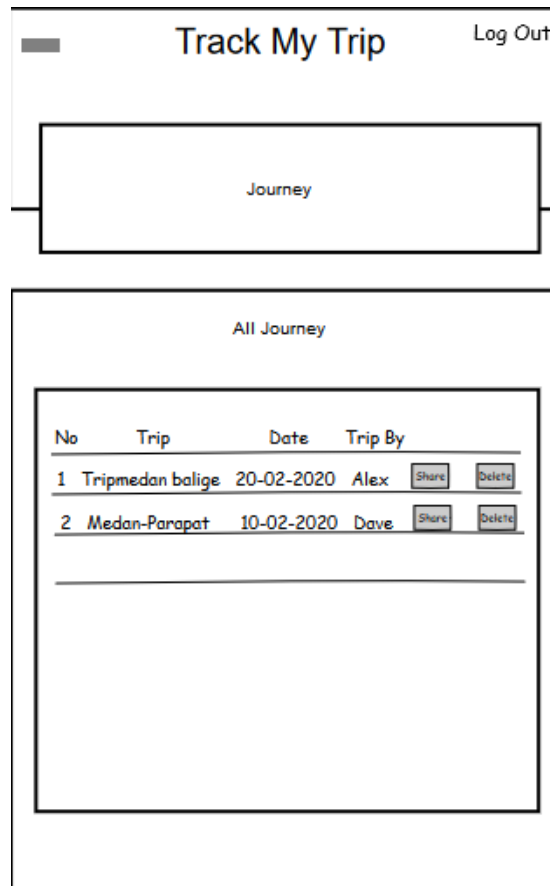


**Gambar 3.47. Design UI Perjalanan**

Pada desain tampilan track journey menampilkan gambaran bagaimana untuk memulai perjalanan dan merekam perjalanan. Setelah selesai direkam, perjalanan pun dapat disimpan.

#### **3.8.3.5.6 Desain *User Interface* Menghapus Data Perjalanan**

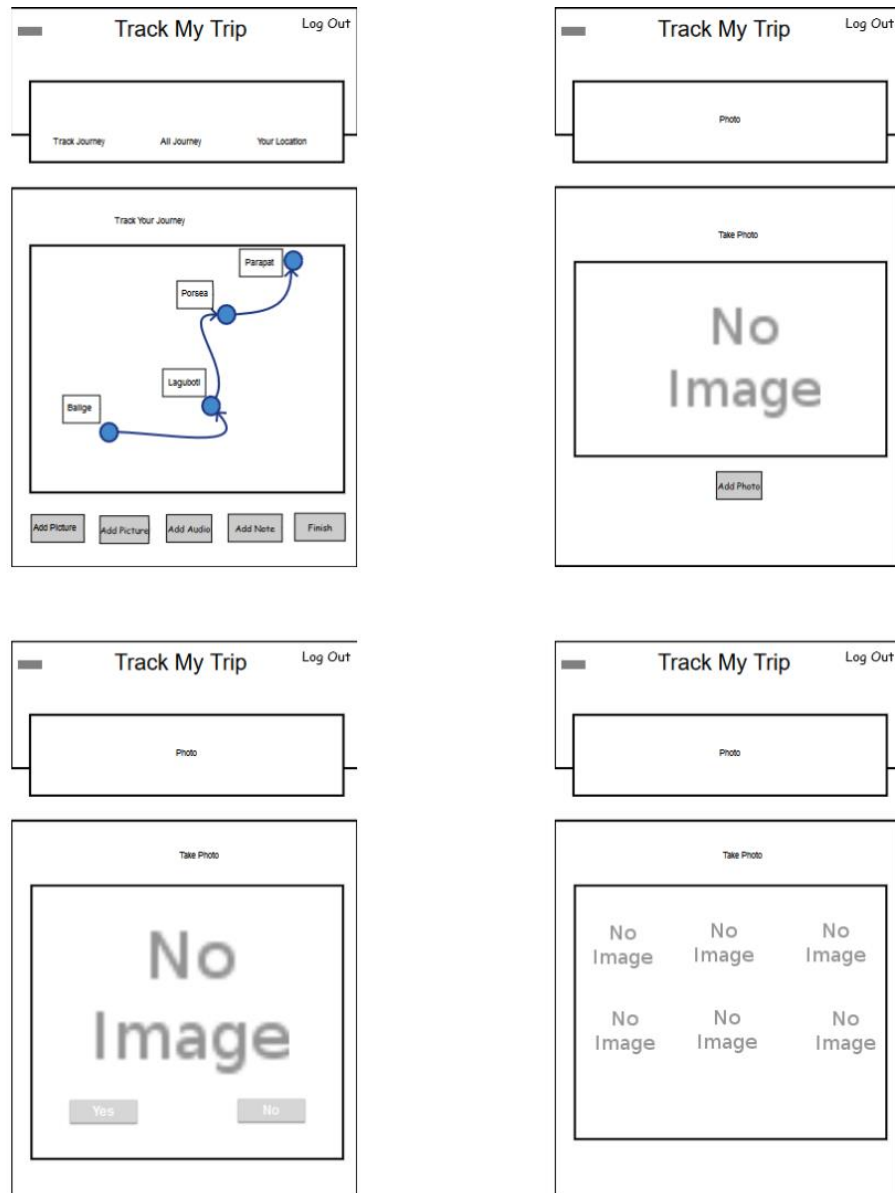
Desain *user interface* menghapus data perjalanan, desain ini menggambarkan fungsional untuk menghapus data perekaman perjalanan oleh pengguna. Pada Gambar 3.48 terdapat desain tampilan menghapus data perjalanan.



Gambar 3.48. Desain UI Menghapus Data Perjalanan

#### 3.8.3.5.7 Desain User Interface Merelasikan Lokasi dengan Foto

Desain *user interface* merelasikan foto dengan lokasi. Desain ini menggambarkan *user interface* fungsi untuk melakukan merelasikan gambar dengan data perjalanan yang dilakukan oleh pengguna, desain *interface* bagaimana data gambar dapat diambil atau pengambilan gambar dan juga bagaimana gambaran desain *interface* bagaimana foto atau data gambar dapat dilihat atau ditampilkan. Pada Gambar 3.49 berikut ini menunjukkan desain *user interface* merelasikan lokasi dengan foto.



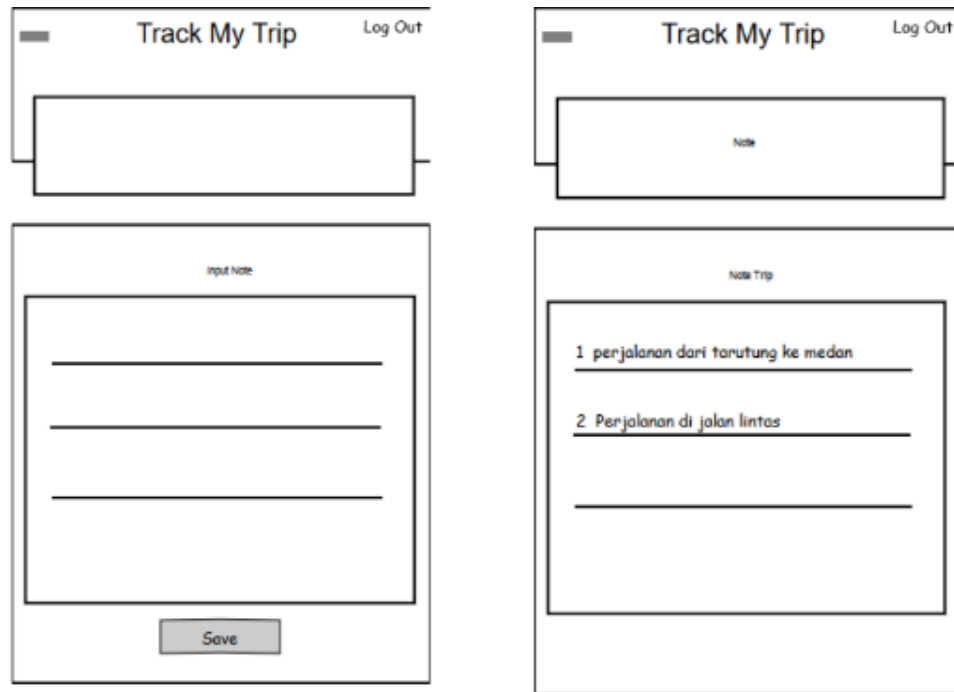
**Gambar 3.49. Design UI Menambahkan Foto**

Gambar diatas merupakan desain untuk mengambil foto dalam aplikasi, setelah foto berhasil diambil, maka akan dikonfirmasi apakah foto akan disimpan.

#### **3.8.3.5.8 Desain User Interface Merelasikan Lokasi dengan Note**

Desain *user interface* merelasikan note dengan lokasi ini menggambarkan desain bagaimana note atau catatan di merelasikan lokasi dengan lokasi atau koordinat, desain bagaimana note dapat ditambahkan ke dalam data perjalanan dan terelasikan,

dan juga desain bagaimana *user* dapat melihat data *note* yang telah direkam atau tersimpan dalam aplikasi. Pada Gambar 3.48 berikut ini menunjukkan desain *user interface* merelasikan lokasi dengan *note*.

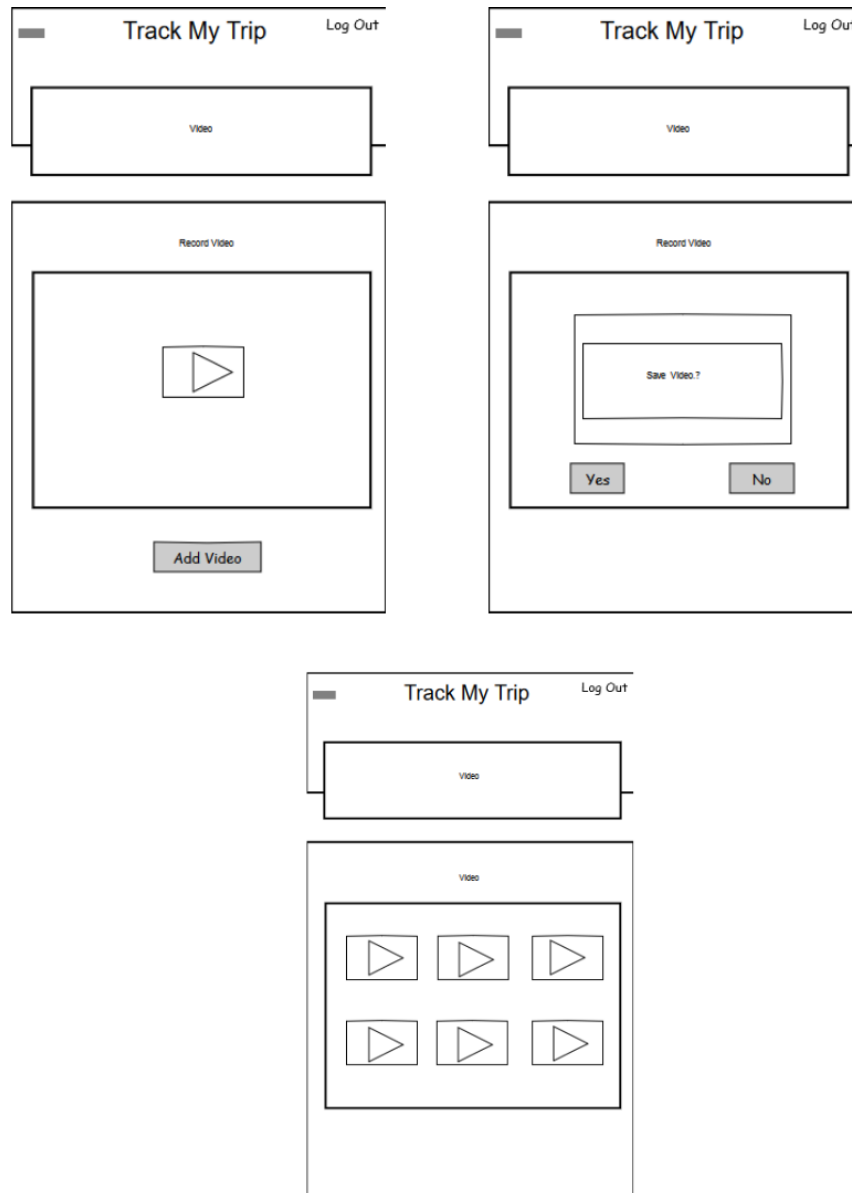


**Gambar 3.50. Design UI Menambahkan Note**

Pada gambar diatas merupakan desain untuk menambahkan catatan, setelah catatan di tambahkan lalu data akan disimpan dan ditambahkan kedalam data perjalanan.

#### **3.8.3.5.9 Desain User Interface Merelasikan Lokasi dengan Video**

Desain *user interface* untuk pengguna merelasikan video dengan lokasi. Dalam desain *user interface* aplikasi ini menggambarkan merelasikan antara lokasi dengan video yang direkam dalam aplikasi, *interface* desain video yang tersimpan dan desain tampilan bagaimana *user* dapat memutar video yang telah tersimpan dalam aplikasi. Pada Gambar 3.51 berikut ini menunjukkan desain *user interface* merelasikan lokasi dengan video.



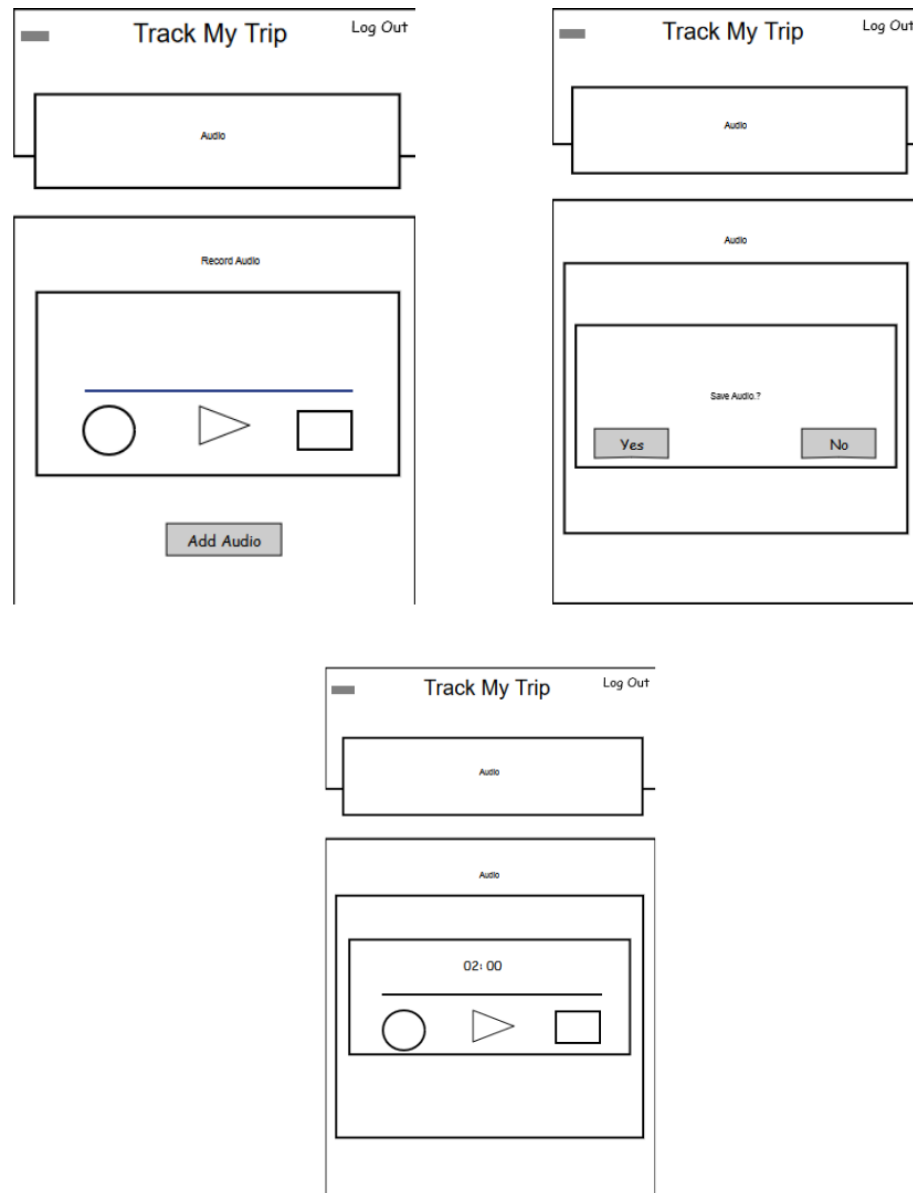
**Gambar 3.51. Design UI Menambahkan Video**

Gambar 3.51 merupakan desain merelasikan video dalam aplikasi, pada desain tersebut digambarkan bagaimana *user* melakukan perekaman video dengan aplikasi, video yang terekam akan tersimpan dalam aplikasi dan juga dapat di putar.

#### **3.8.3.5.10 Desain *User Interface* Merelasikan Lokasi dengan Voice**

Pada desain *user interface* ini menjelaskan fungsi untuk melakukan merelasikan antara rekaman suara dengan *coordinate* perjalanan yang dilakukan oleh *user*, desain ini menggambarkan bagaimana suara atau audio direkam dalam aplikasi dan

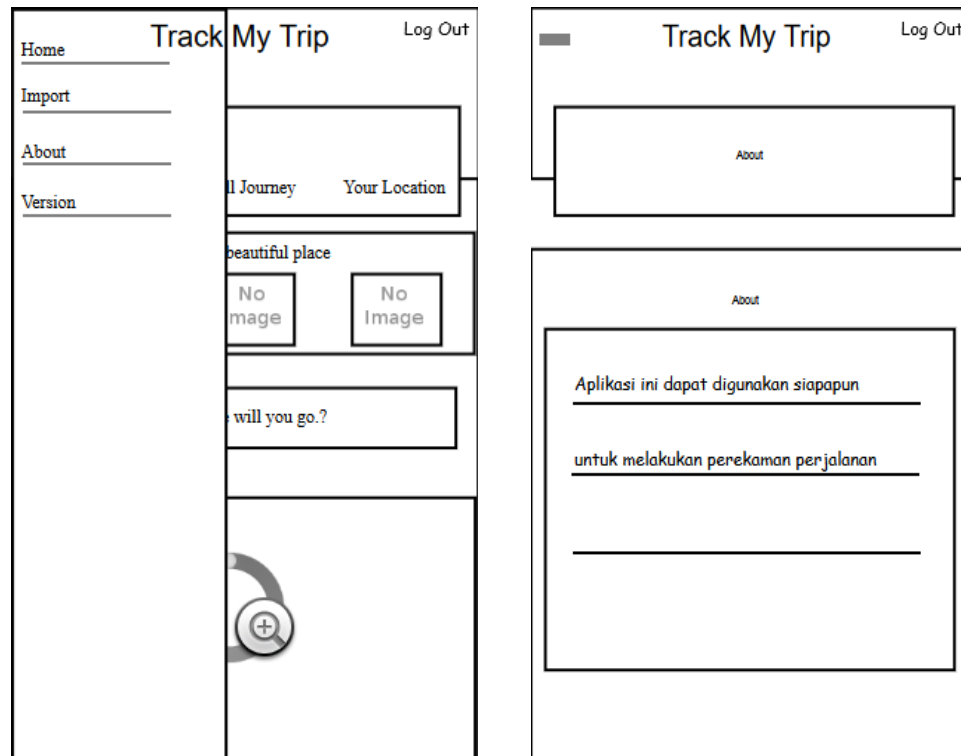
juga menampilkan atau membuka audio yang telah tersimpan. Pada Gambar 3.52 berikut ini menunjukkan desain *user interface* merelasikan lokasi dengan suara.



**Gambar 3.52. Design UI untuk Menambahkan Audio**

#### **3.8.3.5.11 Desain User Interface About**

Desain *user interface* About menggambarkan penjelasan tentang aplikasi, meliputi versi aplikasi, fungsional dan penggunaan aplikasi serta penjelasan gambaran umum aplikasi *track my trip*. Pada Gambar 3.53 terdapat desain about.

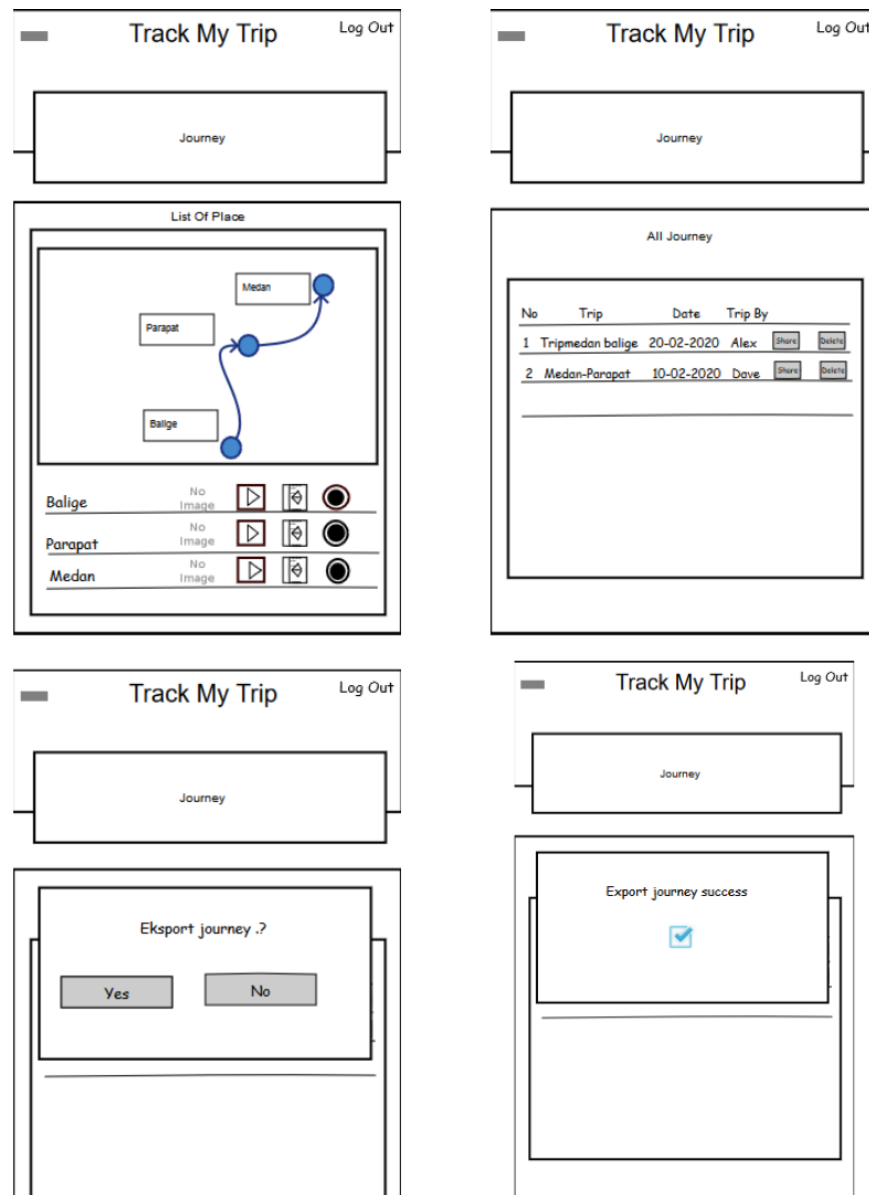


Gambar 3.53. Design UI About

#### 3.8.3.5.12 Desain *User Interface* Share

Desain Gambar 3.54 berikut ini menggambarkan bagaimana membagikan perjalanan yang telah direkam dalam aplikasi, sehingga dengan demikian data perjalanan dapat di share ke pengguna lainnya.





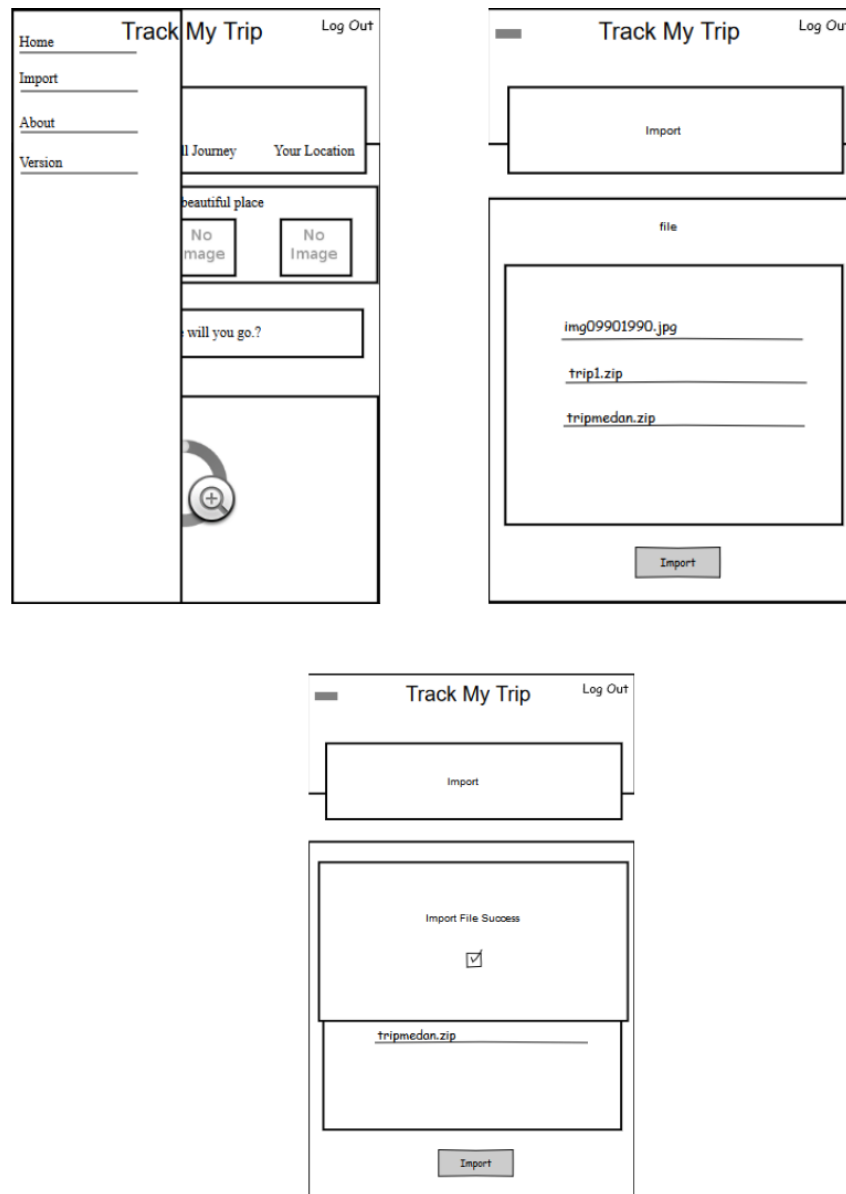
Gambar 3.54. Design User Interface Share perjalanan

Pada desain di atas *user* memilih icon Share pada data perjalanan, kemudian akan muncul pop-up yang mengkonfirmasi apakah data akan di Share, jika *user* memilih 'yes' maka Share akan berhasil dilakukan dan data akan tersimpan ke penyimpanan lokal.

### 3.8.3.5.13 Desain User Interface Impor

Desain pada Gambar 3.55 berikut ini adalah desain antar muka impor data perjalanan pada aplikasi, data yang terdapat dalam aplikasi berupa data perjalanan

yang di Share sebelumnya, selain itu dapat juga diimpor oleh aplikasi atau pengguna lainnya sehingga dapat sama sama melihat perjalanan masing-masing pengguna.



**Gambar 3.55. Design User Interface Impor Data Perjalanan**

Pada desain antarmuka diatas menjelaskan gambaran bagaimana cara mengimpor data perjalanan ke dalam aplikasi, klik icon menu di pojok kiri atas aplikasi, lalu muncul navigasi, lalu pilih menu impor, setelah itu akan muncul form untuk “attach” file perjalanan, setelah itu pilih file yang akan diimpor berupa file Zip perjalanan lainnya, lalu pilih impor dan selesai.

## BAB 4

### IMPLEMENTASI

Pada bab ini menjelaskan mengenai deskripsi umum terkait aplikasi, yang meliputi kebutuhan implementasi, batasan implementasi, implementasi aplikasi dan implementasi fungsi.

#### 4.1 Kebutuhan Implementasi

Pada bagian ini menjelaskan mengenai spesifikasi *software* dan *hardware* yang dibutuhkan dalam pengembangan aplikasi. Spesifikasi perangkat lunak yang digunakan pada pengimplementasian aplikasi dapat dilihat pada Tabel 4.1. Spesifikasi perangkat keras yang digunakan dapat dilihat pada Tabel 4.2.

**Tabel 4.1. Kebutuhan Implementasi pada Software**

Laptop 1	Lenovo G40
Processor	Intel Core i7
RAM	8.00 GB
Hard disk	1TB

Laptop 2	Lenovo IDEAPAD 320
Processor	Intel Core i7
RAM	8.00 GB
Hard disk	1TB

Laptop 3	HP 14 Notebook PC
Processor	Intel(R) Core(TM) i3-4030U
RAM	4.00 GB
Hard disk	500GB

Tabel 4.2. Kebutuhan Implementasi pada Hardware

No.	Software	Specification
1.	<i>Operating System</i>	Windows 10
2.	<i>Development Tools</i>	Android Studio dan Jenkins
3.	<i>Programming Language</i>	Java
4.	<i>Database Tools</i>	Firebase
5.	<i>Framework</i>	
6.	<i>Design Tools</i>	Enterprise Architect, Pensils, Bizagi

#### 4.2 Batasan Implementasi

Aplikasi yang dibangun adalah aplikasi *tracking* untuk semua perjalanan. Pengembangan aplikasi ini memiliki batasan dalam pengimplementasian yaitu :

1. Pengembangan Aplikasi Track My Trip menggunakan Bahasa pemrograman Java untuk aplikasi *mobile*.
2. Aplikasi Track My Trip versi *mobile* digunakan pengguna melihat lokasi keberadaan nya, melakukan perekaman perjalanan, dan merelasikan titik koordinatnya dengan multimedia seperti foto, video, voice record dan note.
3. Pengguna aplikasi Track My Trip dapat membagikan perjalanannya kepada orang lain.

#### 4.3 Persiapan Praktik *Continuous Integration Continuous Delivery*

Pada sub bab akan dijelaskan tentang persiapan yang dilakukan untuk menerapkan praktik *Continuous Integration Continuous Delivery*. Untuk persiapan yang dilakukan antara lain pembuatan *repository* yaitu Gitlab kemudian instalasi Jenkins pada lokal pengembang. Setelah instalasi berhasil, maka dilakukan instalasi *plugin* yang dibutuhkan.

1. Kode *repository* pada Gitlab

Kode *repository* digunakan sebagai wadah untuk para pengembang menggabungkan setiap kode program yang telah ditulis. Kode program yang ditulis kemudian disimpan pada *repository* yang telah dibuat yaitu Gitlab.

## 2. Instalasi Jenkins dan Instalasi *Plugin*

Jenkins adalah *tools* yang digunakan untuk praktik *Continuous Integration*. Jenkins harus diinstal pada lokal pengembang. Setelah Jenkins berhasil diinstal, langkah selanjutnya adalah instalasi *plugin*. Pada TINJAUAN PUSTAKA di SubBab 2.6.2 dijelaskan langkah untuk instalasi Jenkins.

## 3. Konfigurasi Gitlab dengan Jenkins

Sebelum menerapkan *Continuous Integration*, terlebih dahulu dilakukan konfigurasi antara GitLab dengan Jenkins untuk memastikan bahwa antara Gitlab dan Jenkins terkoneksi dan dapat berfungsi dengan baik. Pada TINJAUAN PUSTAKA di SubBab 2.6.2 dijelaskan langkah untuk konfigurasi GitLab dengan Jenkins.

## 4.4 Proses Implementasi Aplikasi Track My Trip

Pada bagian ini dijelaskan mengenai aplikasi yang diimplementasikan dalam pengerjaan Tugas Akhir. Aplikasi Track My Trip yang dibangun merupakan aplikasi yang berbasis *mobile*. Aplikasi yang berbasis *mobile* dikembangkan menggunakan Android Studio dengan bahasa pemrograman Java. Aplikasi ini akan menggunakan *automation tools* yaitu Jenkins. Untuk penyimpanan data akan menggunakan *firebase*.

Pada bagian ini dijelaskan mengenai proses implementasi aplikasi yang diimplementasikan pada aplikasi Track My Trip.

### 4.4.1 Implementasi Fungsi pada Aplikasi Track My Trip

Pada bagian ini dijelaskan mengenai implementasi fungsi yang diimplementasikan pada aplikasi Track My Trip serta input dan output dari setiap fungsi. Berikut ini potongan *code* dari fungsi aplikasi.

#### 4.4.1.1 Implementasi Menu Home

Berikut pada Tabel 4.3 merupakan potongan *code* untuk implementasi menu home aplikasi Track My Trip.

**Tabel 4.3. Potongan Kode Program Implementasi Home**

```

public void backHome(MenuItem item) {
    fragmentManager = getSupportFragmentManager();
    fragmentTransaction = fragmentManager.beginTransaction();
    fragmentTransaction.replace(R.id.nav_host_fragment, new yourLocation());
    fragmentTransaction.commit();
}

```

#### 4.4.1.2 Implementasi Fungsi Autentikasi

Autentikasi akan dilakukan jika sudah masuk ke dalam aplikasi. Pertama, *user* akan register. Setelah akun terdaftar, maka *user* akan login ke dalam aplikasi. Berikut pada Table 4.5 merupakan potongan *code* untuk login. Pada Tabel 4.4 merupakan potongan *code* untuk register.

**Tabel 4.4. Potongan Kode Program Implementasi Register**

```

fAuth.createUserWithEmailAndPassword(email, pass)
    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if(task.isSuccessful()) {
                Toast.makeText(registrasi.this, "User created.",
                    Toast.LENGTH_SHORT).show();
                userID = fAuth.getCurrentUser().getUid();
                DocumentReference documentReference =
                    fStore.collection("user").document(userID);
                documentReference.set(user).addOnSuccessListener(new
                    OnSuccessListener<Void>() {
                        @Override
                        public void onSuccess(Void aVoid) {
                            Log.d(TAG, "onSuccess: user profile is
                                created for " + userID);
                        }
                    });
                startActivity(new
                    Intent(getApplicationContext(), home.class));
            } else {
                Toast.makeText(registrasi.this, "Register
                    Unsuccessfully", Toast.LENGTH_SHORT)
                    .show();
            }
        }
    });
}

```

**Tabel 4.5. Potongan Kode Program Implementasi Login**

```
fAuth.signInWithEmailAndPassword(email,pass).addOnCompleteListener(new
OnCompleteListener<AuthResult>() {
    @Override
    public void onComplete(@NonNull Task<AuthResult> task) {
        if(task.isSuccessful()) {
            Toast.makeText(Login.this, "Login Successfully.",
Toast.LENGTH_SHORT).show();
            startActivity(new Intent(getApplicationContext(),home.class));
        }else{
            Toast.makeText(Login.this, "Login Unsuccessfully",
Toast.LENGTH_SHORT).show();
        }
    }
});
```

#### 4.4.1.3 Implementasi Fungsi Lokasi Pengguna

User bisa melihat titik lokasi dimana pengguna berada. Berikut pada Tabel 4.6 merupakan potongan *code* untuk fungsi lokasi pengguna.

**Tabel 4.6. Potongan Kode Program Implementasi Lokasi Pengguna**

```
private void getCurrentLocation() {
    if (ActivityCompat.checkSelfPermission(getContext(),
Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED &&
        ActivityCompat.checkSelfPermission(getContext(),
Manifest.permission.ACCESS_COARSE_LOCATION) !=
PackageManager.PERMISSION_GRANTED) {
        ActivityCompat.requestPermissions(getActivity(),
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION}, 44);
    }
    Task<Location> task = client.getLastLocation();
    task.addSuccessListener(new OnSuccessListener<Location>() {
        @Override
        public void onSuccess(final Location location) {
            if(location != null){
                supportMapFragment.getMapAsync(new OnMapReadyCallback() {
                    @Override
                    public void onMapReady(GoogleMap googleMap) {
                        LatLng latLng = new LatLng(location.getLatitude(),
location.getLongitude());
                        MarkerOptions options = new
MarkerOptions().position(latLng).title("Im Here");
                        googleMap.animateCamera(CameraUpdateFactory.newLatLngZoom(latLng, 10));
                        googleMap.addMarker(options);
                    }
                });
            }
        }
    });
}
```

```
}  
    } ) ;  
}  
} ) ;  
}
```

#### 4.4.1.4 Implementasi Fungsi Rekam Perjalanan

*User* bisa melakukan perekaman perjalanan ke tempat yang akan dituju. Perekaman akan dilakukan setiap 10 menit. Berikut pada Tabel 4.7 merupakan potongan *code* untuk fungsi rekam perjalanan pengguna.

**Tabel 4.7. Potongan Kode Program Implementasi Rekam Perjalanan**

```

if (ActivityCompat.checkSelfPermission(getContext(),
Manifest.permission.ACCESS_FINE_LOCATION)
    == PackageManager.PERMISSION_GRANTED) {
    timer.schedule(new TimerTask() {
        @Override
        public void run() { // Function runs every MINUTES minutes.
            getCurrentLocation();
        }
    }, 0, 1000 * 60 * MINUTES);
}

```

#### 4.4.1.5 Implementasi Fungsi Merelasikan Lokasi dengan Foto

*User* bisa melakukan pengambilan foto dimana *user* berada. Berikut pada Tabel 4.8 merupakan potongan *code* untuk fungsi merelasikan lokasi dengan foto.

**Tabel 4.8. Potongan Kode Program Implementasi Lokasi dengan Foto**

```
private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    // Ensure that there's a camera activity to handle the intent
    if
    (takePictureIntent.resolveActivity(getActivity().getPackageManager()) !=
    null) {
        // Create the File where the photo should go
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            System.out.println(ex.fillInStackTrace());
        }
    }
}
```



```

    }
    // Continue only if the File was successfully created
    if (photoFile != null) {
        Uri photoURI = FileProvider.getUriForFile(getActivity(),
            "com.example.ta1920ti13",
            photoFile);
        takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
        startActivityForResult(takePictureIntent, CAMERA_REQUEST_CODE);
    }
}
}

```

#### 4.4.1.6 Implementasi Fungsi Merelasikan Lokasi dengan Video

User bisa melakukan pengambilan video dimana *user* berada. Berikut pada Tabel 4.9 merupakan potongan *code* untuk fungsi merelasikan lokasi dengan video.

**Tabel 4.9. Potongan Kode Program Implementasi Lokasi dengan Video**

```

private void dispatchTakeVideoIntent() {
    Intent takeVideoIntent = new Intent(MediaStore.ACTION_VIDEO_CAPTURE);
    if (takeVideoIntent.resolveActivity(getActivity().getPackageManager())
    != null) {
        startActivityForResult(takeVideoIntent, REQUEST_VIDEO_CAPTURE);
    }
}

```

#### 4.4.1.7 Implementasi Fungsi Merelasikan Lokasi dengan Voice Record

User bisa melakukan perekaman suara dimana *user* berada. Berikut pada Tabel 4.10 merupakan potongan *code* untuk fungsi merelasikan lokasi dengan voice record.

**Tabel 4.10. Potongan Kode Program Implementasi Lokasi dengan Voice Record**

```

public void startRecorder(){
    if(isRecording){
        //stop;

    btnStart.setImageDrawable(getResources().getDrawable(R.drawable.start,
    null));

        stopRecording();
        isRecording = false;
    }else{
        //start;
        if(checkPermission()) {

    btnStart.setImageDrawable(getResources().getDrawable(R.drawable.stop,
    null));

            startRecording();
            isRecording = true;

        }
    }
}

private void startRecording() {
    timer.setBase(SystemClock.elapsedRealtime());
    timer.start();
    String recordPath =
getActivity().getExternalFilesDir("/").getAbsolutePath();
    SimpleDateFormat format = new SimpleDateFormat("yyyy_mm_dd_hh_mm_ss",
Locale.CANADA);
    Date now = new Date();
    recordFile = "Recording"+ format.format(now) + ".3gp";
    mediaRecorder = new MediaRecorder();
    mediaRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    mediaRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mediaRecorder.setOutputFile(recordPath+ "/" +recordFile);
    mediaRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);
    try {
        mediaRecorder.prepare();
    } catch (IOException e) {
        e.printStackTrace();
    }
    mediaRecorder.start();
}

private void stopRecording() {
    timer.stop();
    mediaRecorder.stop();
    mediaRecorder.release();
    mediaRecorder = null;
}

```

#### 4.4.1.8 Implementasi Fungsi Merelasikan Lokasi dengan Note

User bisa melakukan pencatatan *note* dimana *user* berada. Berikut pada Tabel 4.11 merupakan potongan *code* untuk fungsi merelasikan lokasi dengan *note*.

**Tabel 4.11. Potongan Kode Program Implementasi Lokasi dengan Note**

```
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_note, container, false);

    saveBtn = view.findViewById(R.id.simpanNote);
    kembaliBtn = view.findViewById(R.id.kembaliLagi);

    judul = view.findViewById(R.id.judulNote);
    isi = view.findViewById(R.id.isiNote);

    dataIsi = judul.getText().toString() + "\n" + isi.getText().toString();
    dataJudul = judul.getText().toString();

    saveBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            writeFile();
        }
    });
    kembaliBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            kembaliKeJourney();
        }
    });

    return view;
}
```

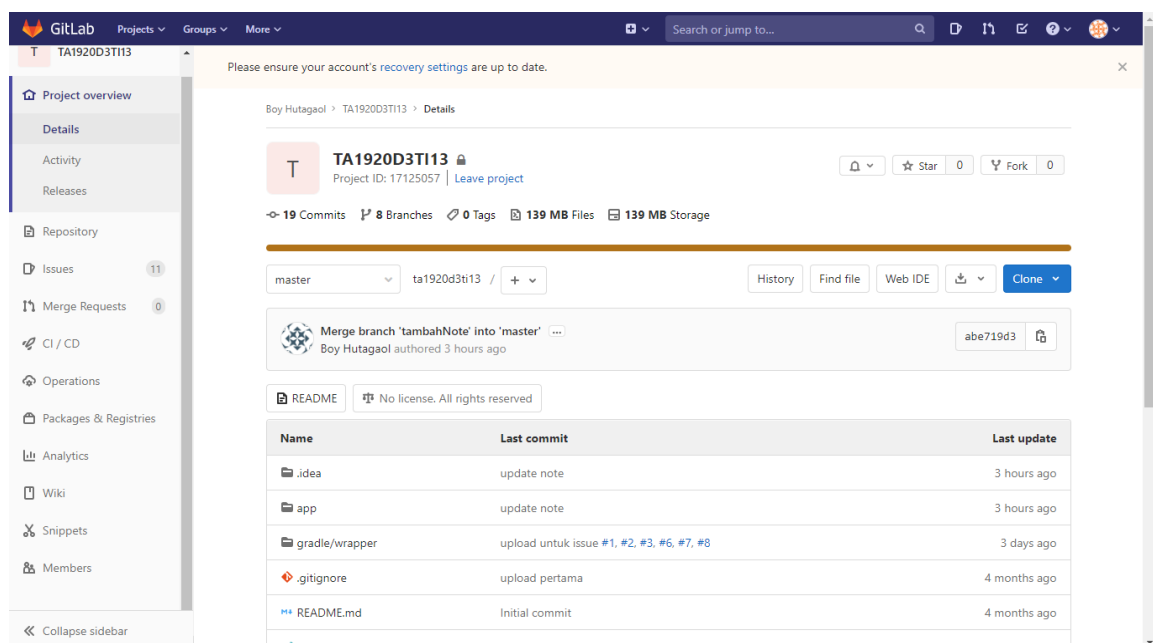
#### 4.4.2 Implementasi Penerapan *Continuous Integration Continuous Delivery*

Pada bagian ini akan dijelaskan penerapan *Continuous Integration*. *Continuous Integration* akan membantu pengembang dalam menyiapkan perangkat lunak agar bisa dirilis ke tahap selanjutnya dan mengecek apakah aplikasi dapat *build* dengan lancar dan tidak ada test yang gagal. Pengembang akan melakukan *push* ke dalam *repository* yang telah dibuat. Jika proses *push* dari pengembang berhasil (*success*) atau tidak error, maka akan lanjut ke proses *build* Jenkins. Proses *build* akan otomatis sesuai dengan jadwal yang ditetapkan.

Penggunaan *tools* pada pengembangan Aplikasi Track My Trip mencakup pengelolaan *repository* dan pengelolaan Jenkins pada proses integrasi.

##### 4.4.2.1 Pengelolaan *Repository*

Pada sub bab ini dijelaskan mengenai pengelolaan *repository*. Pada Tugas Akhir ini akan dibuat suatu *repository* pada Gitlab yang bernama TA1920D3TI13. Kode program yang telah ditulis oleh tim pengembang akan disimpan ke dalam *repository* tersebut untuk menjaga agar versi kode program tetap tetap terkoordinir dan terpantau sehingga tim pengembang mengetahui versi terbaru dan setiap perubahan pada *repository*. Pada Gambar 4.1 dapat dilihat *repository* yang digunakan pada Tugas Akhir ini.



Gambar 4.1. *Repository* pada Gitlab

#### 4.4.2.2 Penerapan CICD dengan Jenkins

Pada sub bab ini akan dijelaskan mengenai proses penerapan *continuous integration continuous delivery* dengan menggunakan Jenkins. Tahapan-tahapan nya dimulai dari *integrated, compiled, tested, archived* dan *deployed*.

##### a. *Integrated*

Integrasi kode program dilakukan dengan menggabungkan semua *functional requirement* dari aplikasi Track My Trip menggunakan Jenkins. Penggunaan Jenkins pada proses integrasi dapat dilakukan dengan mengikuti langkah yang telah dibuat sebelumnya. Jenkins akan mengambil kode program yang berada pada Gitlab dan melakukan *build* otomatis untuk mengeksekusi kode program yang telah diambil sebelumnya dari Gitlab. Tim pengembang melakukan penjadwalan *build* otomatis dengan *build triggers*.

##### b. *Compiled*

*Build Triggers* berfungsi untuk membuat penjadwalan *build* otomatis berjalan. *Schedule* yang berada pada *build periodically* yaitu “00 08 \*\* 0-6” dan “00 22 \*\*0-6” yang berarti jadwal *build* yang ditetapkan untuk pengembangan aplikasi Track My Trip adalah mulai dari hari Minggu sampai dengan hari Sabtu pada pukul 08.00 dan 22.00 WIB. Jadwal *build* dapat dilihat pada Gambar 4.2.

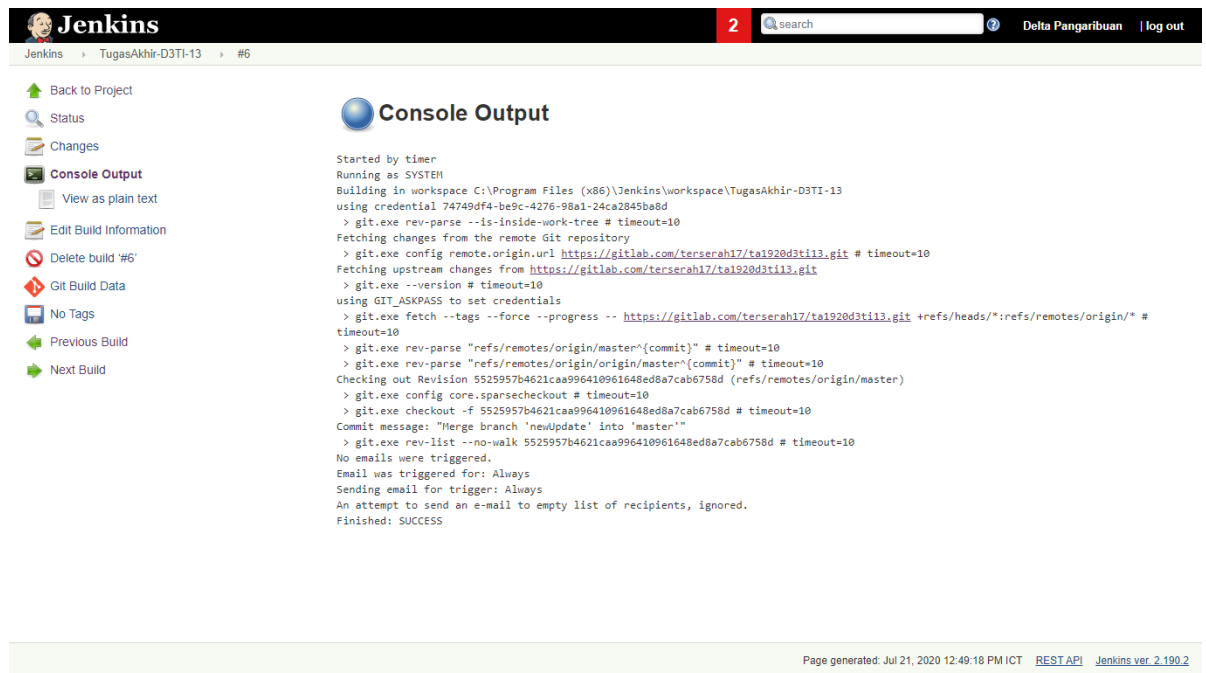
Schedule	00 08 ** 0-6 00 22 ** 0-6
----------	------------------------------

Gambar 4.2. *Build Schedule*

*Build* otomatis akan berjalan sesuai jadwal yang telah ditetapkan dan mengambil perubahan terakhir yang ada pada *repository*.

### c. *Tested*

Pada Gambar 4.3 dapat dilihat hasil *build* otomatis dari perubahan terakhir yang ada pada Gitlab.



**Gambar 4.3. Console Output**

Gambar 4.3 menunjukkan bahwa *build* berhasil (*success*) ditandai dengan warna titik (*dot color*) berubah menjadi biru.

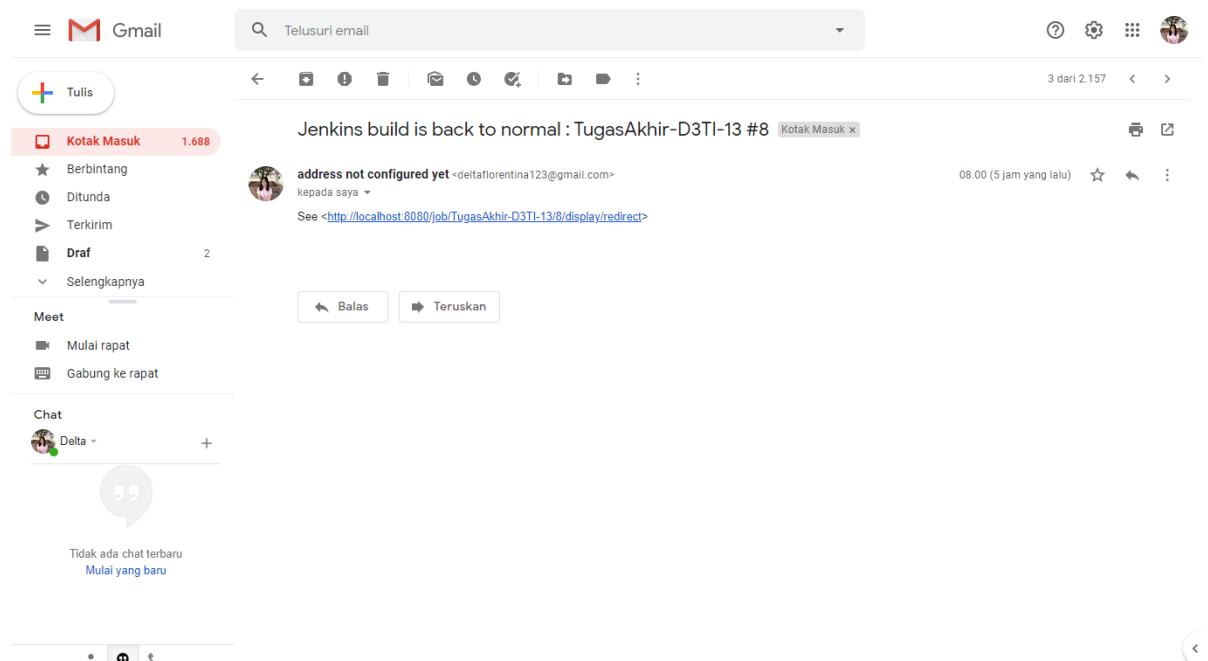
### d. *Archived*

Hasil dari *build* tersebut akan diarsipkan pada halaman utama proyek yaitu TugasAkhir- D3TI-13 yang dibuat pada Jenkins. Pada halaman utama proyek akan terdapat list *build* (*history build*) yang telah dijalankan sebelumnya. Pada list *build* (*history build*) akan terdapat list dengan warna titik (*dot color*) biru dan merah. Jika *build* yang dijalankan berhasil maka warna titik (*dot color*) berubah menjadi biru dan sebaliknya jika *build* yang dijalankan gagal maka warna titik (*dot color*) berubah menjadi merah. Pada Gambar 4.4 dapat dilihat list *build* (*history build*) aplikasi Track My Trip.

Build History <span>trend</span>	
find	x
<b>#10</b>	Jul 22, 2020 8:00 AM
<b>#9</b>	Jul 21, 2020 10:00 PM
<b>#8</b>	Jul 21, 2020 8:00 AM
<b>#7</b>	Jul 20, 2020 10:00 PM
<b>#6</b>	Jul 19, 2020 8:00 AM
<b>#5</b>	Jul 18, 2020 10:00 PM
<b>#4</b>	Jul 18, 2020 8:00 AM

**Gambar 4.4. Build History**

Pada saat *build* telah selesai dilakukan, akan dikirimkan notifikasi berupa *email* ke alamat *email* tim pengembang yang telah ditentukan sebelumnya. Pada Gambar 4.5 akan menunjukkan *email* yang telah dikirimkan ke alamat email yang ditentukan dan *email* berisi hasil *build* yang menunjukkan bahwa *build* ke-8 berhasil (*success*).



**Gambar 4.5. Email Notification**

Pada Tabel 4.12 dapat dilihat *build* yang telah dilakukan sebelumnya :

**Tabel 4.12. Daftar *Build***

Tanggal <i>Build</i>	Hasil <i>Build</i>	<i>Commit Message</i>	<i>Commit SHA</i>
30 Maret 2020	<i>Success</i>	"Merge branch 'halamanUtama'"	ad17463ce62a2db1ecad02e5eb2de760e0380c57
6 April 2020	<i>Success</i>	"Merge branch 'autentikasi2' into 'master'"	f29fc5fcc2bd8f75c9495984fa370d4ca56b581b
6 April 2020	<i>Success</i>	"finalisasi created account"	6f8801bbdb0dc9e0875f41b740746732abb662df
6April 2020		"usercurrentlocation"	376af6a4d76987121aa216d623fd45c3ff55a67d
16 Mei 2020	<i>Success</i>	"perbaiki map"	fa8187af0b9c49ca4f71ab69b9f8320f84a9ee0b
16 Juli 2020	<i>Succes</i>	"issue #1 login"	b635e3a2e4b5396504718a9c37b6c5d05fe89fcf
17 Juli 2020	<i>Success</i>	"Merge branch 'map2' into 'master'"	eb6b4d4ac7cadee9c04b08728ed78c3869aca140
17 Juli 2020	<i>Success</i>	"Merge branch 'map2' into 'master'"	eb6b4d4ac7cadee9c04b08728ed78c3869aca140
17 Juli 2020	<i>Success</i>	"Merge branch 'map2' into 'master'"	eb6b4d4ac7cadee9c04b08728ed78c3869aca140
18 Juli 2020	<i>Success</i>	"Merge branch 'map2' into 'master'"	eb6b4d4ac7cadee9c04b08728ed78c3869aca140



<b>Tanggal Build</b>	<b>Hasil Build</b>	<b>Commit Message</b>	<b>Commit SHA</b>
18 Juli 2020	<i>Success</i>	"Merge branch 'newUpdate' into 'master'"	5525957b4621caa996410961648ed8a7cab6758d
18 Juli 2020	<i>Success</i>	"Merge branch 'newUpdate' into 'master'"	5525957b4621caa996410961648ed8a7cab6758d
18 Juli 2020	<i>Failed</i>		
18 Juli 2020	<i>Success</i>	"Merge branch 'newUpdate' into 'master'"	5525957b4621caa996410961648ed8a7cab6758d
18 Juli 2020	<i>Success</i>	"Merge branch 'newUpdate' into 'master'"	5525957b4621caa996410961648ed8a7cab6758d
18 Juli 2020	<i>Success</i>	"Merge branch 'newUpdate' into 'master'"	5525957b4621caa996410961648ed8a7cab6758d
18 Juli 2020	<i>Success</i>	"Merge branch 'updateTrackJourney' into 'master'"	722f3dde87c7c97bde46daef9bfc641ecfc49593

## **BAB 5**

### **PENGUJIAN**

Pada bab ini akan dijelaskan skenario pengujian aplikasi Track My Trip yang meliputi hal yang diuji pada versi *mobile*. Tujuan dilakukannya pengujian adalah untuk memastikan fungsi yang dibangun telah berjalan sesuai dengan rancangan dan berjalan dengan baik dan benar.

#### **5.1 Tujuan Pengujian**

Tujuan dari pengujian terhadap aplikasi yang dibangun adalah sebagai berikut:

1. Mengujicobakan, memeriksa, mengecek dan memastikan apakah fungsi yang telah dibuat pada aplikasi dapat berjalan dengan baik sesuai dengan perancangan yang sebelumnya telah dilakukan pada tahap perancangan.
2. Pengujian ini untuk memverifikasi, memvalidasi data dan mendeteksi apakah terdapat kesalahan pada aplikasi.
3. Memperoleh hasil *feedback* dari aplikasi yang telah dilakukan pengujian, untuk dapat menyelesaikan permasalahan pada aplikasi.

#### **5.2 Skenario Pengujian Aplikasi**

Pada proses pengujian yang dilakukan terhadap aplikasi Track My Trip terdiri dari dua tahapan, yaitu :

1. Pengujian aplikasi oleh tim pengembang
2. Pengujian aplikasi oleh *user*

Berikut ini merupakan skenario uji dan hasil uji yang dilakukan pada aplikasi Track My Trip:

### 5.2.1 BU-01 Autentikasi

Pengujian autentikasi dilakukan untuk memastikan *user* mendaftarkan akun nya ke dalam lokasi dan masuk ke dalam aplikasi sebelum menggunakan aplikasi. Skenario pengujian autentikasi dapat dilihat pada Tabel 5.1.

**Tabel 5.1. Butir Uji Autentikasi**

Identifikasi	BU-01		
No. Fungsi	FR01		
Nama Butir Uji	Pengujian autentikasi		
Tujuan	Uji coba fungsi autentikasi		
Deskripsi	Fungsi ini dijalankan oleh pengguna untuk menyimpan data <i>user</i> agar bisa masuk ke dalam aplikasi.		
Kondisi Awal	1. Aplikasi di- <i>install</i>		
Skenario Pengujian			
1. Mengisi semua field pada <i>form</i> registrasi			
2. Mengisi field pada <i>form</i> registrasi dengan data yang salah			
3. Menekan <i>button</i> registrasi			
4. Mengisi semua field pada <i>form</i> login			
5. Mengisi field pada <i>form</i> login dengan data yang salah			
6. Menekan <i>button</i> login			
Kriteria Evaluasi Hasil			
Ketika <i>form</i> registrasi telah di <i>create</i> maka data registrasi dapat masuk ke database dan akun pun berhasil dibuat dan akan masuk ke dalam <i>form</i> login. Tetapi, ketika field pada <i>form</i> registrasi diisi dengan data yang salah, misalnya alamat email yang dimasukkan tidak benar maka akan muncul alert “Alamat email yang dimasukkan salah”. Ketika akun berhasil dibuat dan masuk ke dalam <i>form</i> login, maka <i>form</i> login di <i>create</i> maka data dapat masuk ke dalam database dan pengguna dapat masuk ke dalam aplikasi.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

• Pengguna memasukkan alamat email sebagai <i>username</i> , nomor handphone, dan <i>password</i> untuk akun yang telah didaftarkan.	Pengguna berhasil mendaftarkan akun nya ke dalam aplikasi	Aplikasi menampilkan data registrasi	Berhasil
• Pengguna memasukkan <i>username</i> dan <i>password</i> yang telah didaftarkan.	Pengguna berhasil login ke dalam aplikasi	Aplikasi menampilkan data login	Berhasil
<b>Catatan</b>			
Apabila muncul alert misalnya “Alamat email yang dimasukkan salah” maka yang harus dilakukan adalah mengisi alamat email dengan benar beserta data yang lainnya.			

### 5.2.2 BU-02 Menunjukkan Lokasi Pengguna

Pengujian menunjukkan lokasi pengguna dilakukan untuk memastikan *user* bisa melihat keberadaan lokasi dimana *user* berada. Skenario pengujian menunjukkan lokasi pengguna dapat dilihat pada Tabel 5.2.

**Tabel 5.2. Butir Uji Menunjukkan Lokasi Pengguna**

<b>Identifikasi</b>	BU-02
<b>No. Fungsi</b>	FR02
<b>Nama Butir Uji</b>	Pengujian menunjukkan lokasi pengguna
<b>Tujuan</b>	Uji coba fungsi menunjukkan lokasi pengguna
<b>Deskripsi</b>	Fungsi ini dijalankan oleh <i>user</i> untuk menunjukkan lokasi keberadaan <i>user</i>
<b>Kondisi Awal</b>	3. Login ke sistem 4. Masuk ke halaman home
<b>Skenario Pengujian</b>	
1. Tidak mengaktifkan GPS	

Kriteria Evaluasi Hasil			
Ketika <i>user</i> telah login ke dalam aplikasi dan telah mengaktifkan GPS, maka <i>user</i> dapat melihat lokasi keberadaannya pada home. Tetapi, pada saat GPS tidak diaktifkan maka lokasi dari pengguna tidak terlihat.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	<i>User</i> dapat melihat lokasi dimana pengguna tersebut berada	Aplikasi menampilkan peta lokasi pengguna	Berhasil
Catatan			

### 5.2.3 BU-03 Rekam Perjalanan

Pengujian rekam perjalanan dilakukan untuk memastikan *user* bisa melakukan perekaman perjalanan nya. Skenario pengujian rekam perjalanan pengguna dapat dilihat pada Tabel 5.3.

**Tabel 5.3. Butir Uji Rekam Perjalanan**

Identifikasi	BU-03		
No. Fungsi	FR03		
Nama Butir Uji	Pengujian rekam perjalanan		
Tujuan	Uji coba fungsi rekam perjalanan		
Deskripsi	Fungsi ini dijalankan oleh <i>user</i> untuk melakukan perekaman perjalanan <i>user</i>		
Kondisi Awal	<div><div>1. Login ke dalam aplikasi</div><div>2. Masuk ke halaman home</div><div>3. Memilih menu Track Journey</div></div>		
Skenario Pengujian			
<div>1. Menekan <i>button</i> 'start' pada aplikasi</div>			
Kriteria Evaluasi Hasil			
User berhasil melakukan perekaman suatu perjalanan yang ingin dilaluinya.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

	User bisa merekam perjalanan yang ingin direkam	Aplikasi dapat merekam perjalanan	Berhasil
<b>Catatan</b>			

#### 5.2.4 BU-04 Menghapus Data Perjalanan

Pengujian menghapus data perjalanan dilakukan untuk memastikan *user* bisa menghapus data perjalanan yang telah direkam. Skenario pengujian menghapus data perjalanan pengguna dapat dilihat pada Tabel 5.4.

**Tabel 5.4. Butir Uji Menghapus Data Perjalanan**

Identifikasi	BU-04		
No. Fungsi	FR04		
Nama Butir Uji	Menghapus data perjalanan		
Tujuan	Uji coba fungsi menghapus data perjalanan		
Deskripsi	Fungsi ini dijalankan oleh <i>user</i> untuk menghapus data perjalanan yang telah direkam		
Kondisi Awal	1. Login ke aplikasi 2. Masuk ke dalam aplikasi 3. Memilih menu All Journey		
Skenario Pengujian			
1. Melihat data perjalanan 2. Menekan <i>button</i> delete			
Kriteria Evaluasi Hasil			
<i>User</i> berhasil menghapus data perjalanan yang telah direkamnya.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	<i>User</i> bisa menghapus data perjalanan yang telah direkam nya dan dalam aplikasi data perjalanan yang dihapus telah terhapus	Data perjalanan yang ingin dihapus oleh <i>user</i> berhasil terhapus	Berhasil

### 5.2.5 BU-05 Merelasikan Lokasi dengan Foto

Pengujian merelasikan foto dengan lokasi pengguna dilakukan untuk memastikan *user* bisa melakukan pengambilan foto dan direlasikan dengan titik pengambilan foto. Skenario pengujian merelasikan lokasi dengan foto dapat dilihat pada Tabel 5.5.

**Tabel 5.5. Butir Uji Merelasikan Lokasi dengan Foto**

Identifikasi	BU-05		
No. Fungsi	FR05		
Nama Butir Uji	Pengujian merelasikan lokasi dengan foto		
Tujuan	Uji coba fungsi merelasikan lokasi dengan foto		
Deskripsi	Fungsi ini dijalankan oleh <i>user</i> untuk merelasikan lokasi dengan foto		
Kondisi Awal	1. Login ke dalam aplikasi		
Skenario Pengujian			
1. Memilih <i>button track journey</i>			
2. Memilih <i>button</i> kamera			
3. <i>User</i> mengambil gambar			
4. <i>User</i> menyimpan gambar			
Kriteria Evaluasi Hasil			
<i>User</i> berhasil melakukan pengambilan foto dan menyimpan nya dalam aplikasi.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	<i>User</i> bisa mengambil foto dan menyimpannya dalam aplikasi sesuai dengan lokasi pengambilan foto	Aplikasi berhasil membuka kamera dan menyimpan gambar	Berhasil
Catatan			

### 5.2.6 BU-06 Merelasikan Lokasi dengan Video

Pengujian merelasikan video dengan lokasi pengguna dilakukan untuk memastikan *user* bisa melakukan pengambilan video dan direlasikan dengan titik pengambilan

video. Skenario pengujian merelasikan lokasi dengan video dapat dilihat pada Tabel 5.6.

**Tabel 5.6. Butir Uji Merelasikan Lokasi dengan Video**

Identifikasi	BU-06		
No. Fungsi	FR06		
Nama Butir Uji	Pengujian merelasikan lokasi dengan video		
Tujuan	Uji coba fungsi merelasikan lokasi dengan video		
Deskripsi	Fungsi ini dijalankan oleh <i>user</i> untuk merelasikan lokasi dengan video		
Kondisi Awal	1. Login ke dalam aplikasi		
Skenario Pengujian			
1. Memilih <i>button track journey</i>			
2. Memilih <i>button video</i>			
3. <i>User</i> melakukan perekaman video			
4. <i>User</i> menyimpan rekaman video			
Kriteria Evaluasi Hasil			
<i>User</i> berhasil melakukan pengambilan video dan menyimpan nya ke dalam aplikasi.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	<i>User</i> bisa mengambil video dan menyimpannya dalam aplikasi sesuai dengan lokasi pengambilan video	Aplikasi berhasil membuka kamera dan merekam video	Berhasil
Catatan			

### 5.2.7 BU-07 Merelasikan Lokasi dengan Voice Record

Pengujian merelasikan voice record dengan lokasi pengguna dilakukan untuk memastikan *user* bisa melakukan perekaman suara dan direlasikan dengan titik perekaman suara. Skenario pengujian merelasikan lokasi dengan voice record dapat dilihat pada Tabel 5.7.



Tabel 5.7. Butir Uji Merelasikan Lokasi dengan Voice Record

Identifikasi	BU-07		
No. Fungsi	FR07		
Nama Butir Uji	Pengujian merelasikan lokasi dengan voice record		
Tujuan	Uji coba fungsi merelasikan lokasi dengan voice record		
Deskripsi	Fungsi ini dijalankan oleh <i>user</i> untuk merelasikan lokasi dengan voice record		
Kondisi Awal	1. Login ke dalam aplikasi		
Skenario Pengujian			
1. Memilih <i>button track journey</i>			
2. Memilih <i>button record</i>			
3. <i>User</i> melakukan perekaman suara			
4. <i>User</i> menyimpan data rekaman suara			
Kriteria Evaluasi Hasil			
<i>User</i> berhasil melakukan perekaman voice record dan menyimpan nya ke dalam aplikasi.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	<i>User</i> bisa merekam suaranya dan menyimpannya dalam aplikasi sesuai dengan lokasi perekaman suara	Aplikasi berhasil merekam suara dan menyimpan suara	Berhasil
Catatan			

### 5.2.8 BU-08 Merelasikan Lokasi dengan Note

Pengujian merelasikan note dengan lokasi pengguna dilakukan untuk memastikan *user* bisa melakukan pencatatan note dan direlasikan dengan titik pengambilan note. Skenario pengujian merelasikan lokasi dengan note dapat dilihat pada Tabel 5.8.

Tabel 5.8. Butir Uji Merelasikan Lokasi dengan Note

<b>Identifikasi</b>	BU-08
<b>No. Fungsi</b>	FR08
<b>Nama Butir Uji</b>	Pengujian merelasikan lokasi dengan note
<b>Tujuan</b>	Uji coba fungsi merelasikan lokasi dengan note

Deskripsi	Fungsi ini dijalankan oleh <i>user</i> untuk merelasikan lokasi dengan voice record		
Kondisi Awal	1. Login ke dalam aplikasi		
Skenario Pengujian			
1. Memilih <i>button track journey</i> 2. Memilih <i>button note</i> 3. <i>User</i> membuat catatan selama perjalanan 4. <i>User</i> dapat menyimpan note			
Kriteria Evaluasi Hasil			
<i>User</i> berhasil melakukan pencatatan note dan menyimpan nya ke dalam aplikasi.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	<i>User</i> bisa menulis <i>note</i> dan menyimpannya dalam aplikasi sesuai dengan lokasi penulisan <i>note</i>	Aplikasi berhasil membuka note dan menyimpan note	Berhasil
Catatan			

### 5.2.9 BU-09 Share Perjalanan

Pengujian share perjalanan pengguna dilakukan untuk memastikan *user* bisa membagikan perjalanan kepada *user* lain. Skenario pengujian *share* perjalanan dapat dilihat pada Tabel 5.9.

**Tabel 5.9. Butir Uji *Share* Perjalanan**

<b>Identifikasi</b>	BU-09
<b>No. Fungsi</b>	FR09
<b>Nama Butir Uji</b>	Pengujian <i>share</i> perjalanan
<b>Tujuan</b>	Uji coba fungsi <i>share</i> perjalanan
<b>Deskripsi</b>	Fungsi ini dijalankan oleh <i>user</i> untuk <i>share</i> perjalanan atau membagikan data perjalanannya

Kondisi Awal	<div><div>1. Login ke dalam aplikasi</div><div>2. User menyimpan data perjalanan yang telah direkam</div><div>3. Membuka data file perjalanan yang tersimpan</div><div>4. Memilih Data perjalanan yang akan di bagikan</div><div>5. Pilih dan bagikan data perjalanan atau share</div></div>		
Skenario Pengujian			
User dapat membagikan dokumentasi dari data perjalanan yang sudah dilakukan oleh user dengan pengguna lainnya			
Kriteria Evaluasi Hasil			
User berhasil membagikan data perjalanan nya kepada user lain.			
Kasus dan Hasil Pengujian			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
	User bisa membagikan data perjalanan kepada user lain		Berhasil
Catatan			

### 5.2.10 BU-10 Impor Data Perjalanan

Pengujian impor data perjalanan pengguna dilakukan untuk memastikan *user* bisa mengimpor. Skenario pengujian impor data perjalanan dapat dilihat pada Tabel 5.10.

**Tabel 5.10. Butir Uji Impor Data Perjalanan**

<b>Identifikasi</b>	BU-10
<b>No. Fungsi</b>	FR10
<b>Nama Butir Uji</b>	Pengujian impor data perjalanan
<b>Tujuan</b>	Uji coba fungsi impor data perjalanan
<b>Deskripsi</b>	Fungsi ini dijalankan oleh pengguna untuk impor data perjalanan
<b>Kondisi Awal</b>	1. Login ke dalam aplikasi 2. Memilih menu impor data perjalanan 3. Mengambil data perjalanan dari penyimpanan lokal smartphone 4. Memilih dan mengimport data yang di pilih user
<b>Skenario Pengujian</b>	

User dapat mengimport data perjalanan dari penyimpanan lokal kedalam aplikasi secara langsung.			
<b>Kriteria Evaluasi Hasil</b>			
<i>User</i> berhasil melakukan import data perjalanannya.			
<b>Kasus dan Hasil Pengujian</b>			
<b>Data Masukan</b>	<b>Yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
	<i>User</i> bisa mengimpor data perjalanan nya		Berhasil
<b>Catatan</b>			

## BAB 6

### HASIL DAN PEMBAHASAN

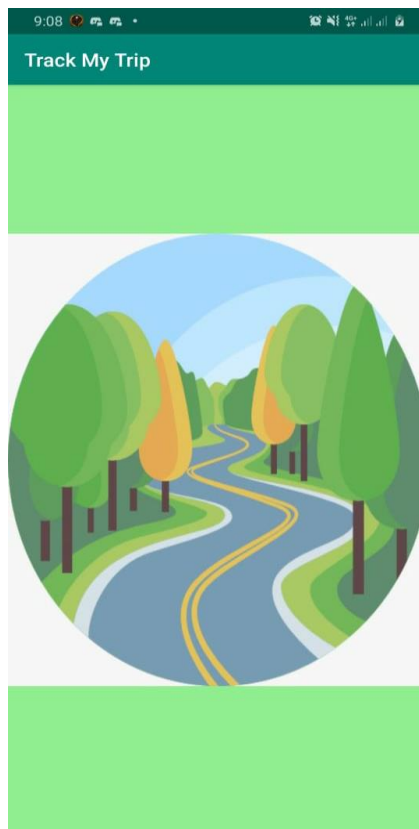
Pada bab ini menjelaskan bagaimana hasil dan pembahasan dari implementasi aplikasi Track My Trip versi *mobile* menerapkan konsep CICD.

#### 6.2 Hasil Aplikasi

Aplikasi yang telah dihasilkan adalah aplikasi *tracking* untuk melakukan perekaman semua jenis perjalanan dalam aplikasi mobile. Aplikasi tersebut dapat merelasikan perjalanannya dengan multimedia seperti foto, video, note dan voice record dari pengguna.

##### 6.2.1 Splashscreen

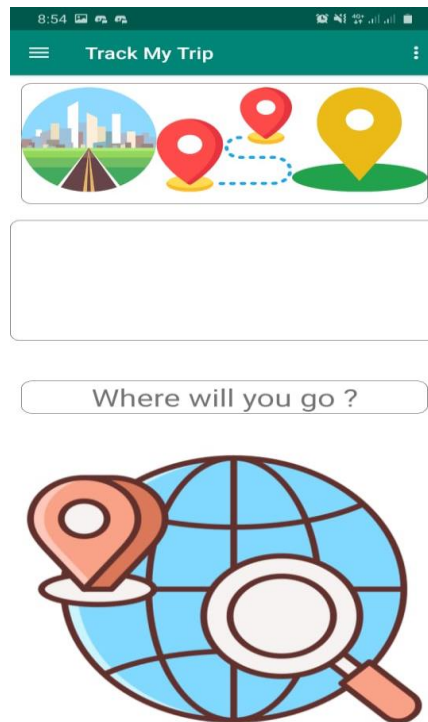
Tampilan Splashscreen dari aplikasi Track My Trip dapat dilihat pada Gambar 6.1.



Gambar 6.1. Tampilan Splashscreen

### 6.2.2 Home

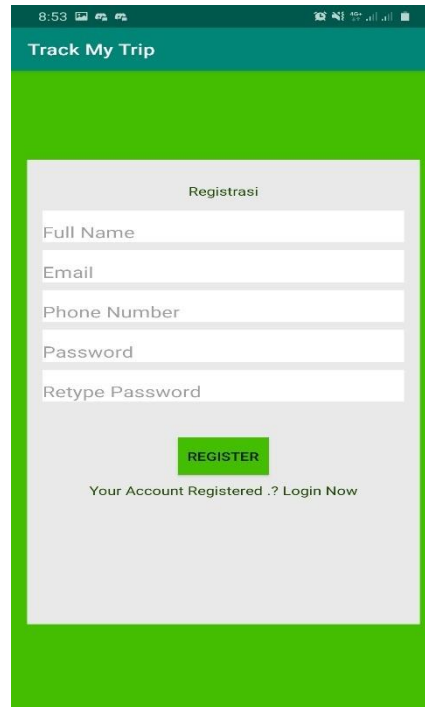
Tampilan Home dari aplikasi Track My Trip dapat dilihat pada Gambar 6.2.



**Gambar 6.2. Tampilan Home**

### 6.2.3 Autentikasi

Tampilan Register dari aplikasi Track My Trip dapat dilihat pada Gambar 6.3 dan tampilan Login dari aplikasi dapat dilihat pada Gambar 6.4.



8:53

Track My Trip

Registrasi

Full Name

Email

Phone Number

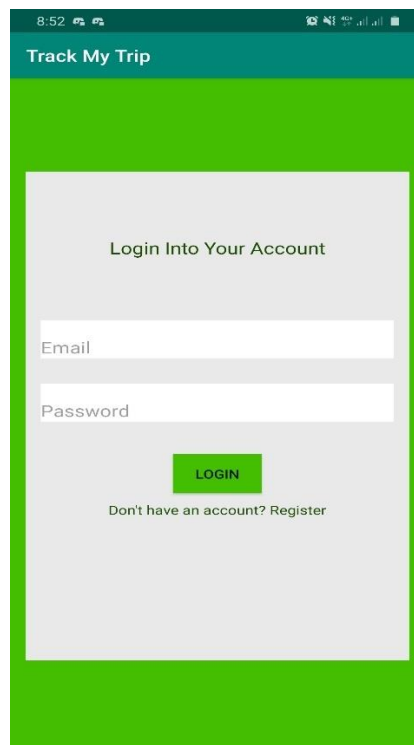
Password

Retype Password

REGISTER

Your Account Registered .? Login Now

**Gambar 6.3. Tampilan Register**



8:52

Track My Trip

Login Into Your Account

Email

Password

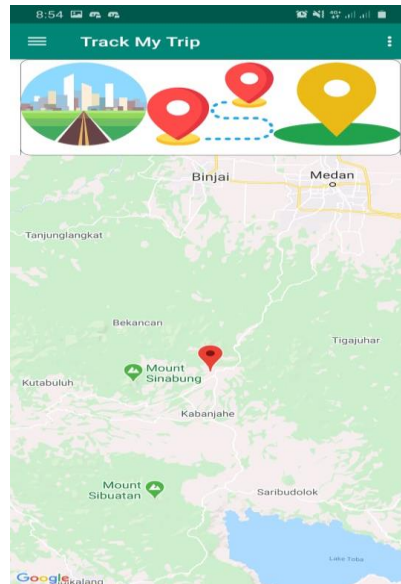
LOGIN

Don't have an account? Register

**Gambar 6.4. Tampilan Login**

### 6.2.4 Lokasi Pengguna

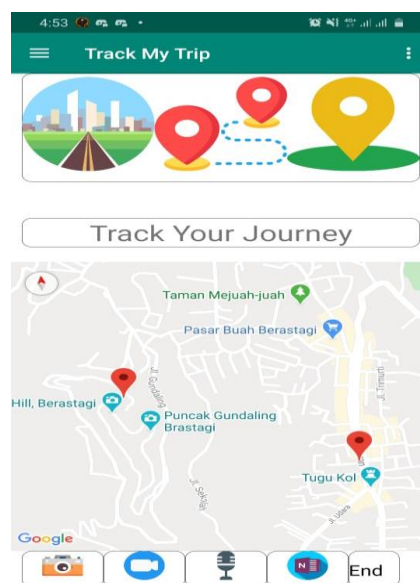
Tampilan Lokasi Pengguna dari aplikasi Track My Trip dapat dilihat pada Gambar 6.5.



**Gambar 6.5. Tampilan Lokasi Pengguna**

### 6.2.5 Rekaman Perjalanan

Tampilan Rekaman Perjalanan dari aplikasi Track My Trip dapat dilihat pada Gambar 6.6.

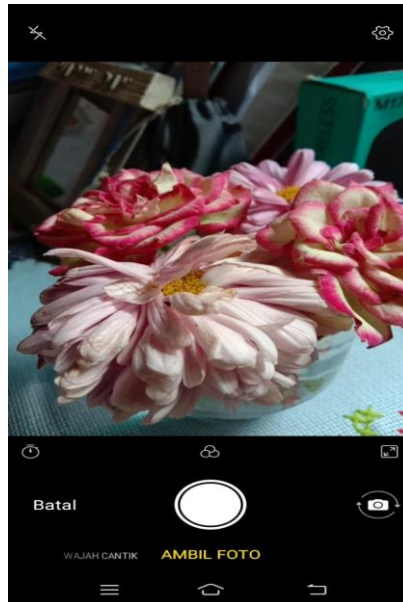


**Gambar 6.6. Tampilan Track Journey**



### 6.2.6 Foto dengan Lokasi

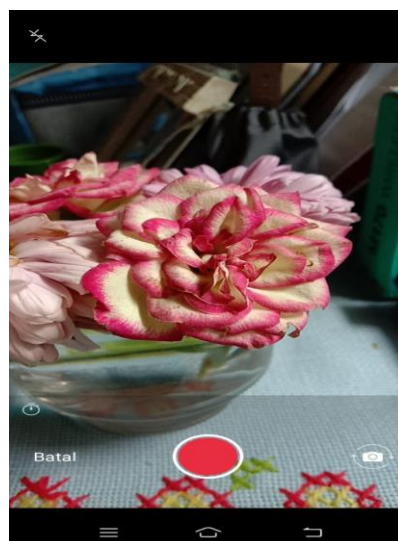
Tampilan Foto dengan Lokasi dari aplikasi Track My Trip dapat dilihat pada Gambar 6.8.



Gambar 6.7. Tampilan Ambil Gambar

### 6.2.7 Video dengan Lokasi

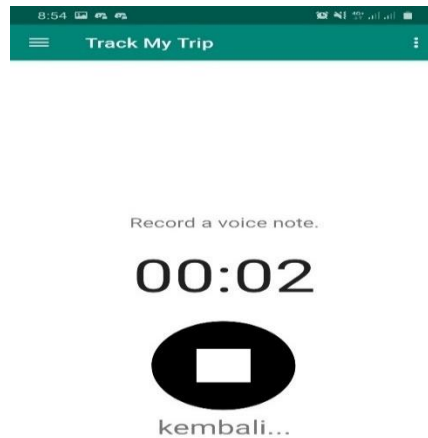
Tampilan Video dengan Lokasi dari aplikasi Track My Trip dapat dilihat pada Gambar 6.9.



Gambar 6.8. Tampilan Rekam Video

### 6.2.8 Voice record dengan Lokasi

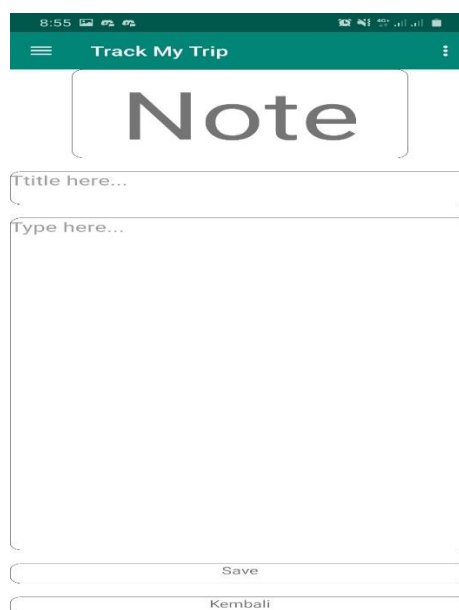
Tampilan Voice record dengan Lokasi dari aplikasi Track My Trip dapat dilihat pada Gambar 6.9.



**Gambar 6.9. Tampilan Record Suara**

### 6.2.9 Note dengan Lokasi

Tampilan Note dengan Lokasi dari aplikasi Track My Trip dapat dilihat pada Gambar 6.10.



**Gambar 6.10. Tampilan Note**

### 6.3 Pembahasan

Aplikasi yang dibangun merupakan aplikasi *tracking* perjalanan yang tidak dibatasi pada suatu perjalanan, melainkan dapat dipakai untuk berbagai jenis perjalanan seperti pariwisata, perjalanan survey lokasi atau tempat untuk kepentingan tertentu seperti lokasi pembangunan jalan, gedung, dll. Fitur aplikasi *tracking* yang ada pada aplikasi Track My Trip memiliki kesamaan dengan fitur yang terdapat pada aplikasi *tracking* pada umumnya. Dan ditambah dengan fitur untuk merelasikan lokasi dengan multimedia seperti foto, video, voice record dan note. Aplikasi Track My Trip ini ada dalam versi *mobile*. Pada pengembangan aplikasi menerapkan konsep CICD pada saat implementasi. Prasyarat untuk penerapan CICD tersebut dibutuhkan *repository* sebagai wadah penyatuan kode program dan *automation tools* untuk menyiapkan perangkat lunak agar bisa dirilis ke tahap selanjutnya dan mengecek apakah aplikasi dapat *build* dengan lancar dan tidak ada test yang gagal. Untuk *repository* yang digunakan adalah Gitlab dan *automation tools* yang digunakan adalah Jenkins. Konsep ini memberikan kemudahan kepada tim pengembangan, mulai dari merelasikan kode antara tim pengembang dan mendapatkan *feedback* apakah mengalami kegagalan atau berhasil pada saat *build*. Terdapat perbedaan antara pengembangan perangkat lunak dengan menggunakan CICD dan tanpa menggunakan CICD. Perbandingan tersebut dapat dilihat pada Tabel 6.1.

**Tabel 6.1. Perbandingan Pengembangan Perangkat Lunak dengan CICD dan Tanpa CICD**

Kriteria	Menerapkan CICD	Tidak Menerapkan CICD
Waktu pengembangan (integrasi kode program)	Kurang efisien karena tim pengembang harus fokus ke pengembangan semua kebutuhan	Efisien, karena tim pengembang fokus pada satu kebutuhan
Waktu pengujian	Kurang efisien karena pengujian dilakukan terhadap keseluruhan sistem	Efisien, karena pengujian dilakukan setiap satu commit selesai dilakukan

Kriteria	Menerapkan CICD	Tidak Menerapkan CICD
Frekuensi commit (integrasi kode program)	Jarang	Sering
Jumlah <i>bugs</i> (integrasi kode program)	Banyak, karena pengujian dilakukan pada keseluruhan sistem	Sedikit, karena setiap fitur diuji dan diperbaiki setelah selesai, tidak perlu menunggu keseluruhan sistem selesai
Biaya	Biaya adalah waktu dan pengerjaan yang dilakukan secara manual dalam pengembangan dan integrasi perangkat lunak	Biaya didapat dari server dan alat yang digunakan dalam CICD

## **BAB 7**

### **KESIMPULAN DAN SARAN**

Pada bab ini berisi kesimpulan dan saran dari pengerjaan Tugas Akhir ini yang diperoleh selama proses pengerjaan Tugas Akhir dan saran yang diperoleh untuk pengembangan Tugas Akhir di masa mendatang.

#### **7.1 Kesimpulan**

Adapun kesimpulan yang telah diperoleh selama pengerjaan Tugas Akhir ini adalah:

1. Aplikasi Track My Trip dapat digunakan seseorang untuk menyimpan atau mendokumentasikan suatu perjalanan yang telah dilalui dalam bentuk multimedia yaitu foto, video, voice dan note.
2. Dalam pengembangan aplikasi Track My Trip menggunakan konsep CICD pada saat tahap implementasi pengembangan perangkat lunak dapat meningkatkan efisiensi dalam pengembangan aplikasi Track My Trip.

#### **7.2 Saran**

Berdasarkan hasil pengerjaan Tugas Akhir ini, saran yang dapat diperhatikan untuk pengembangan Tugas Akhir selanjutnya adalah sebagai berikut:

1. Untuk pengembangan yang akan datang dapat mengembangkan fitur pada aplikasi ini seperti penanda lokasi atau tempat tertentu seperti penginapan, lokasi wisata, tempat kuliner dan tempat-tempat populer di suatu daerah.
2. Komitmen dan disiplin dari tim pengembang harus tetap diperhatikan agar penerapan konsep CICD dapat berjalan dengan baik dan benar.
3. Dalam Tugas Akhir ini data disimpan di eksternal storage menggunakan *firebase* dan penggunaannya harus memiliki akses internet, untuk pengembangan selanjutnya penyimpanan data dapat menggunakan *internal storage* dan *eksternal storage* secara bersamaan, agar penyimpanan data dapat dilakukan baik saat pengguna memiliki akses internet dan saat *offline*.

## Daftar Pustaka

- [1] S. Darekar, A. Chikane, R. Diwate, A. Deshmukh, and A. Shinde, "Tracking System using GPS and GSM: Practical Approach," *Int. J. Sci. Eng. Res.*, vol. 3, no. 5, pp. 3–6, 2012, [Online]. Available: <https://www.ijser.org/researchpaper/tracking-system-using-gps-and-gsm-practical-approach.pdf>.
- [2] A. Juansyah, "Pembangunan Aplikasi Child Tracker Berbasis Assisted – Global Positioning System ( A-GPS ) Dengan Platform Android," *J. Ilm. Komput. dan Inform.*, vol. 1, no. 1, pp. 1–8, 2015, [Online]. Available: [elib.unikom.ac.id/download.php?id=300375](http://elib.unikom.ac.id/download.php?id=300375).
- [3] M. K. Hoq, "Mobile Tracking System using Web Application and Android Apps," vol. 6, no. 02, pp. 257–262, 2017.
- [4] H. P. Yosephat Suryo Susilo, "Sistem Pelacakan Dan Pengamanan Kendaraan Berbasis GPS Dengan Menggunakan Komunikasi GPRS," vol. 13, no. 1, pp. 21–32, 2014.
- [5] R. Ramadi, *Making travel history travel GPS TRACKER Web based on mobile phone using J2ME. Informatics Engineering. University Of Islam State Syarif Hidayatullah Jakarta*. 2011.
- [6] K. C. Dewi, P. I. Ciptayani, and I. W. R. Wijaya, "Agile Project Management Pada Pengembangan E-Musrenbang Kelurahan Benoa Bali Agile Project Management on E-Musrenbang Development in Benoa Village Bali," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 5, no. 6, pp. 723–730, 2018, doi: 10.25126/jtiik.201851143.
- [7] K. M. D. Pertiwi, A. T. Ni'mah, and S. Rochimah, "Survei Dampak Penggunaan Integrasi Berkelanjutan dalam Perusahaan Pengembangan Perangkat Lunak," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 8, no. 2, p. 135, 2019, doi: 10.22146/jnteti.v8i2.502.
- [8] H. Bhanupriya, A. N. Krishna, and R. S. Jha, "Automating Infrastructure as a Code using Continuous Integration and Continuous Delivery," pp. 82–88.
- [9] N. Seth and R. Khare, "ACI (automated Continuous Integration) using Jenkins: Key for successful embedded Software development," *2015 2nd*

- Int. Conf. Recent Adv. Eng. Comput. Sci. RAECS 2015*, no. December, 2016, doi: 10.1109/RAECS.2015.7453279.
- [10] A. Sunyoto, “Deteksi Gempa Dengan Piezo Elektrik Berbasis Mikrokontroler,” pp. 7–33, 2013.
  - [11] S. Komputer and J. Teknik, “Aplikasi interaktif menggunakan sensor pada smartphone berbasis android untuk pembelajaran fisika skripsi,” 2016.
  - [12] S. Dodit and A. Rini, *Pemrograman Aplikasi Android*, no. May. 2013.
  - [13] A. dan S. P. Imaduddin, *Menjadi Android Developer Expert*, I. Bandung, 2017.
  - [14] N. B. Thakkar, “Google Android: An Emerging Innovative Software Platform For Mobile Devices,” *IJIRST –International J. Innov. Res. Sci. Technol.*, vol. 1, no. 6, pp. 272–278, 2014.
  - [15] L. A. Sandy, R. J. Akbar, and R. R. Hariadi, “Rancang Bangun Aplikasi Chat pada Platform Android dengan Media Input Berupa Canvas dan Shareable Canvas untuk Bekerja dalam Satu Canvas Secara Online,” *J. Tek. ITS*, vol. 6, no. 2, 2017, doi: 10.12962/j23373539.v6i2.23782.
  - [16] A. Sonita and R. F. Fardianitama, “Aplikasi E-Order Menggunakan Firebase dan Algoritme Knuth Morris Pratt Berbasis Android,” *Pseudocode*, vol. 5, no. 2, pp. 38–45, 2018, doi: 10.33369/pseudocode.5.2.38-45.
  - [17] M. Faya and N. Fiftin, “Pemanfaatan Google Maps API Untuk Pembangunan Sistem Informasi Manajemen Bantuan Logistik Pasca Bencana Alam Berbasis Mobile Web,” *J. Sarj. Tek. Inform.*, vol. 1, no. 1, pp. 162–171, 2013.
  - [18] “CI/CD: Software yang Selalu Siap untuk Diproduksi oleh Pivotal,” 2018. <http://www.phintraco.com/ci-cd-software-yang-selalu-siap-untuk-diproduksi-oleh-pivotal/>.
  - [19] A. G. Paul M. Duvall, Steve Matyas, *Continuous Integration Improving Software Quality and Reducing Risk*. 2007.
  - [20] R. T. G. Jaga My Pintera, “Automatic Deployment System Dengan Menggunakan Metode Continuous Integration Di Kakatu.”
  - [21] U. Lestari and S. Kristiyana, “Rancang Bangun Mobile Tracking

Application Module Untuk Pencarian Posisi Benda Bergerak Berbasis Short Message Service ( Sms ),” *Semin. Nas. Teknol. Inf. dan Komputasi (SENASTIK 2013)*, vol., no., pp. 30–31, 2013.