

RANCANG BANGUN GPS *BACK TRACK* PADA REKAMAN RUTE PENDAKIAN MENGGUNAKAN SISTEM *EMBEDDED*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Agung Prasetyo
NIM: 135150301111134



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

PENGESAHAN

RANCANG BANGUN GPS BACK TRACK PADA REKAMAN RUTE PENDAKIAN
MENGUNAKAN SISTEM EMBEDDED

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :
Agung Prasetyo
NIM: 135150301111134

Penelitian ini telah diuji dan dinyatakan lulus pada
17 Desember 2018
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dahnial Syaury, S.T., M.T., M.Sc.
NIK: 2016078704231002

Dosen Pembimbing II

Wijaya Kurniawan, S.T., M.T.
NIP: 19820125 201504 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Titi Astoto Kurniawan, S.T., M.T., Ph.D
NIP. 19710518 200312 1 001

PERNYATAAN ORISINILITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah penelitian ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis di sitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah penelitian ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia penelitian ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 17 Desember 2018

Agung Prasetyo

NIM: 135150301111134



KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas limpahan rahmat dan petunjuk-Nya, sehingga penulis dapat menyelesaikan skripsi yang berjudul **“Rancang Bangun GPS Back Track Pada Rekaman Rute Pendakian Menggunakan Sistem Embedded”**. Skripsi ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer di Fakultas Ilmu Komputer Universitas Brawajaya Malang.

Penyusunan dan penulisan skripsi ini tidak lepas dari bantuan moral dan materiil yang diberikan dari berbagai pihak, maka penulis ingin menyampaikan rasa hormat serta mengucapkan terima kasih penulis kepada:

1. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
3. Bapak Dahnil Syauqy, S.T., M.T., M.Sc. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang serta selaku dosen pembimbing I yang telah memberikan pengarahan dan bimbingan kepada penulis, sehingga dapat menyelesaikan penulisan skripsi ini dengan baik.
4. Bapak Wijaya Kurniawan, S.T., M.T., selaku Dosen Pembimbing II yang telah memberikan pengarahan dan bimbingan kepada penulis, sehingga dapat menyelesaikan penulisan skripsi ini dengan baik.
5. Ayahanda bapak Suwaji dan Ibunda Titik Rumiati atas segala nasehat, kasih sayang, perhatian, dan kesabarannya dalam memberikan semangat kepada penulis, serta tiada henti memberikan doa demi terselesaikannya penulisan skripsi ini.
6. Kakak dan keluarga tersayang, Airlangga Hanung Satria Dinarta, Yudi Nuryanta, S.T., Elly Ernawati, S.Pd., atas dukungan semangat dan doa yang diberikan sehingga penulis dapat menyelesaikan penulisan skripsi ini.
7. Riska Amelia Septyarini, S.T., yang penulis sayangi dan banggakan atas doa waktu, perhatian, dukungan, semangat, motivasi, kesabaran serta kasih sayang yang diberikan kepada penulis selama studi hingga pengerjaan skripsi ini dapat terselesaikan.
8. Sahabat Musa Al-Jabar, Dewan Risky Bahari, S.Kom., Imbarotul Muafiq, S.T., M.T., Dwi Putra Ramadhani, S.T., Misbachul Akbar, S.Kom., M.M.S.I., Muhammad Wafi, S.Kom., Fathurozi, S.Kom., Aulia Rizky Pratama, S.Kom., Ahhmadin Imanullah, Ardiyansyah Berok, Yusuf Ucup, Fajarudin, anang, Hanif Sujatmiko, Joko Satria, Bangkit, Irwan Yudha, keluarga kost Joyosuko 61.A dan seluruh sahabat-sahabat penulis atas rasa kebersamaan dan kekeluargaan yang selama ini terjalin serta dukungan, Saran, masukan kepada penulis sehingga penulis dapat menyelesaikan penulisan skripsi ini.
9. Seluruh teman Paguyuban Karya Salemba Empat Universitas Brawijaya yang telah memberi pengalaman, pelajaran, pendidikan dan rasa kekeluargaan selama mengabdikan di Yayasan Karya Salemba Empat.

10. Alumni *XL Future Leaders batch 5* atas pelajaran, pendidikan, kekeluargaan selama ini.
11. Seluruh civitas akademika Informatika Universitas Brawijaya dan terkhusus untuk teman-teman Teknik Komputer 2013 yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Teknik Komputer Universitas Brawijaya dan selama penyelesaian penulisan skripsi ini.
12. Seluruh pihak yang membantu penelitian, penulis berterimakasih atas segala bentuk dukungan dan doa sehingga laporan skripsi ini dapat terselesaikan.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap penelitian ini dapat membawa manfaat bagi semua pihak yang membutuhkannya.

Malang, 26 November 2018

Agung Prasetyo
agungprastt@gmail.com



ABSTRAK

Agung Prasetyo, Rancang Bangun GPS *Back Track* Pada Rekaman Rute Pendakian Menggunakan Sistem *Embedded*.

Pembimbing: Dahnia Syauqy, S.T., M.T., M.Sc., Wijaya Kurniawan, S.T., M.T

Mendaki gunung adalah salah satu kegiatan eksplorasi alam yang membutuhkan pengetahuan dalam memahami peta dan medan yang akan dilalui. Namun, tidak semua pendaki memahami hal tersebut sehingga menyebabkan resiko tersesat saat melakukan kegiatan pendakian. Pada umumnya alat bantu navigasi yang sudah ada hanya mampu memandu pendaki dari rute awal sampai tujuan akhir dan tidak dapat menyimpan dan merekam rute perjalanan yang telah dilewati, sehingga pendaki kesulitan untuk kembali ke rute awal perjalanan pada saat tersesat. Oleh sebab itu, dalam meminimalisir resiko tersebut diperlukan sebuah teknologi alat bantu navigasi yang mampu menampilkan kembali rute perjalanan yang dilewati. Berdasarkan permasalahan tersebut maka dalam penelitian ini dirancang sebuah teknologi alat bantu navigasi yang bernama GPS *back track*. Alat ini dirancang menggunakan beberapa komponen yakni, modul GPS U-Blox Neo 6 M yang berfungsi dalam pengambilan data GPS berupa titik koordinat *latitude* dan *longitude*, LCD *display* 800 x 480 TFT berfungsi menampilkan informasi hasil dan proses perekaman rute yang sedang berlangsung. Raspberry pi 2B berfungsi mengolah data *input* dan *output* selama proses perekaman rute dan sumber daya utama dalam menjalankan seluruh sistem menggunakan *power bank*. GPS *back track* mempunyai fitur yakni dapat membuka file hasil rekaman yang sudah dibuat dan ditampilkan pada LCD *display*. Berdasarkan hasil pengujian akurasi GPS yang dilakukan diketahui persentase error modul GPS pada koordinat *latitude* sebesar 3.77505% dan *longitude* sebesar 3.698758%. Sedangkan dalam pengujian perekaman alat, fungsi *track* dan *trackback* dapat merekam rute perjalanan serta menampilkannya pada LCD berupa garis rute yang dilalui.

Kata kunci : *Back Track, GPS, Navigasi, Raspberry pi 2B, Tracking*

ABSTRACT

Mountain climbing is one of nature's exploration activities that requires knowledge in understanding the map and area to be passed. However, not all climbers have knowledge of navigation, its causing the risk of getting lost while climbing. Navigation assistant that exist, only guide climbers only first route to final destination and can't save the route that already passed, until it causes climber difficulty going back to first route when they lost. Because that, in minimizing te risk of lost, be required navigation assistant device that cloud show the route has been passed. Based on these problems in this research created a navigation device assistant technology called GPS back track. This device is built with several components such a, the U-Blox Neo 6 M GPS module that functions in retrieving GPS data in the form of latitude and longitude coordinates, LCD display 800 x 480 TFT function displays information in the recording process running routes, Raspberry pi 2B functions process data input and output during the process of recording the route and the main resources for running the entire system using power bank. GPS back track has a feature that is able to open the recorded file that has been created and displayed on the LCD display. Based on the results of GPS accuracy testing, it is known that the GPS module error percentage at latitude coordinates is 3.77505% and longitude is 3.698758%. While in the testing of device recording, tracks and trackbacks functions can record routes and display them on the LCD in the form of route lines that are passed.

Keywords: Back Track, GPS, Navigasi, Raspberry pi 2B, Tracking

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINILITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah	3
1.3 Tujuan.....	3
1.4 Manfaat.....	3
1.5 Batasan masalah	4
1.6 Sistematika pembahasan	4
BAB 2 LANDASAN KEPUSTAKAAN	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori	7
2.2.1 Navigasi	8
2.2.2 GPS	8
2.2.3 Raspberry Pi 2B	10
2.2.4 Display LCD.....	11
2.2.5 Bahasa Pemrograman Python.....	12
2.2.6 Library Modul.....	13
2.2.7 Kalman Filter	13
BAB 3 METODOLOGI	15
3.1 Studi Literatur	15
3.2 Analisis Kebutuhan Sistem.....	16

3.3 Perancangan Sistem	16
3.4 Implementasi Sistem.....	17
3.5 Pengujian dan Analisis	17
3.6 Kesimpulan dan Saran.....	17
BAB 4 REKAYASA KEBUTUHAN.....	19
4.1 Gambaran Umum Sistem.....	19
4.2 Identifikasi Pengguna.....	19
4.3 Kebutuhan Fungsional	20
4.4 Kebutuhan Perangkat Keras.....	20
4.5 Kebutuhan Perangkat Lunak.....	21
4.6 Batasan Desain Sistem	21
BAB 5 PERANCANGAN DAN IMPLEMENTASI	23
5.1 Perancangan.....	23
5.1.1 Perancangan Perangkat Keras	23
5.1.2 Perancangan Perangkat Lunak.....	27
5.2 Implementasi	33
5.2.1 Implementasi Desain sistem GPS <i>Track Back</i>	33
5.2.2 Implementasi Perangkat Lunak.....	35
BAB 6 PENGUJIAN DAN ANALISIS	44
6.1 Pengujian Nilai Akurasi Modul GPS Terhadap Nilai <i>Google Maps</i>	44
6.1.1 Tujuan Pengujian.....	44
6.1.2 Prosedur Pengujian.....	44
6.1.3 Hasil dan Analisis Akurasi Nilai <i>Latitude</i>	45
6.1.4 Hasil dan Analisis Akurasi Nilai <i>Longitude</i>	46
6.1.5 Hasil dan Analisis Modul GPS Terhadap Waktu akuisisi Data.....	47
6.2 Pengujian dan Analisis Terhadap Proses Perekaman Rute.....	47
6.2.1 Tujuan pengujian	48
6.2.2 Prosedur pengujian	48
6.2.3 Hasil dan Analisa Proses Perekaman Rute.....	48
6.3 Pengujian dan Analisis Kinerja Fungsi Open File.....	49
6.3.1 Tujuan Pengujian.....	49

6.3.2	Prosedur Pengujian	49
6.3.3	Hasil dan Analisis Proses Open File Rekaman.....	49
6.4	Pengujian dan Analisis Nilai <i>tracking</i> dengan Kalman Filter.....	50
6.4.1	Tujuan Pengujian.....	50
6.4.2	Prosedur Pengujian	50
6.4.3	Hasil dan Analisis Selisih Nilai Koordianat	51
6.5	Pengujian dan Analisis <i>Tracking</i> Data	51
6.6	Kalman Filter	51
6.6.1	Tujuan Pengujian.....	52
6.6.2	Prosedur Pengujian	52
6.6.3	Hasil dan Analisa <i>Tracking</i> Data Menggunakan Kalman Filter ...	52
6.7	Pengujian dan Analisis perekaman dengan acuan Google maps	53
6.7.1	Tujuan Pengujian.....	53
6.7.2	Prosedur Pengujian	53
6.7.3	Hasil dan Analisis Proses Open File Rekaman.....	53
BAB 7	KESIMPULAN DAN SARAN	55
7.1	Kesimpulan.....	55
7.2	Saran	56
DAFTAR PUSTAKA	57

DAFTAR TABEL

Tabel 2.1 Persamaan dan perbandingan	6
Tabel 2.2 Spesifikasi Ublox Neo 6M Modul	9
Tabel 2.3 Spesifikasi Raspberyy Pi 3	10
Tabel 2.4 Spesifikasi <i>Display LCD</i>	12
Tabel 5.1 Komunikasi pin GPIO	24
Tabel 5.2 Pin komunikasi modul GPS.....	27
Tabel 5.3 <i>Library</i>	35
Tabel 5.4 Fungsi <i>Button Click Track</i>	36
Tabel 5.5 Fungsi <i>Run Tracking</i>	36
Tabel 5.6 Fungsi <i>Button Click TrackBack</i>	38
Tabel 5.7 Fungsi <i>Run Track Back</i>	38
Tabel 5.8 Fungsi <i>Ploting Data</i>	40
Tabel 5.9 Implementasi Kalman Filter	41
Tabel 6.1 Percobaan Modul GPS Dengan <i>Google Maps (Latitude)</i>	45
Tabel 6.2 Percobaan Modul GPS Dengan <i>Google Maps (Longitude)</i>	46
Tabel 6.3 Percobaan Modul GPS Terhadap Waktu Akuisisi Data	47
Tabel 6.4 Selisih Koordinat Hasil <i>Tracking</i>	51

DAFTAR GAMBAR

Gambar 2.1 Satelit GPS	9
Gambar 2.2 Fisik GPS Neo 6 M	9
Gambar 2.3 Fisik <i>Raspberry Pi 2B</i>	10
Gambar 2.4 Fisik LCD (800x480) <i>Touch Display for Raspberry Pi 2B</i>	11
Gambar 2.5 kalman filter	14
Gambar 3.1 Diagram Alir Metodologi Penelitian	15
Gambar 3.2 Diagram Blok Sistem	17
Gambar 5.1 Diagram Blok Perancangan perangkat keras	23
Gambar 5.2 Perancangan perangkat keras	24
Gambar 5.3 Desain Kemasan Sistem	25
Gambar 5.4 Desain perancangan <i>display</i> LCD	26
Gambar 5.5 Perancangan rangkaian modul GPS	27
Gambar 5.6 Diagram <i>Aktivity</i> Sistem	28
Gambar 5.7 <i>Flowchart Ploting Rute Track</i>	29
Gambar 5.8 <i>Flowchart Ploting Rute Trackback</i>	30
Gambar 5.9 <i>Flowchart Sub-Program Tracking Data GPS</i>	31
Gambar 5.10 <i>Flowchart Sub-program Save Data</i>	32
Gambar 5.11 <i>Flowchart Sub-Program Kalman Filter</i>	32
Gambar 5.12 Implementasi Desain GPS <i>Track Back</i>	33
Gambar 5.13 Implementasi Rangkaian Display LCD 800 x 480	34
Gambar 5.14 Implementasi Modul Ublox Neo 6M	34
Gambar 6.1 Hasil Proses Perekaman	48
Gambar 6.2 Hasil Pengujian Fungsi <i>Open File</i>	50
Gambar 6.3 Hasil Tracking Data GPS	52
Gambar 6.4 Hasil Pengujian <i>Track dan Back Track</i> Dengan Acuan <i>Google Maps</i>	54

RANCANG BANGUN GPS *BACK TRACK* PADA REKAMAN RUTE PENDAKIAN MENGGUNAKAN SISTEM *EMBEDDED*

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Agung Prasetyo
NIM: 135150301111134



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab landasan kepustakaan yaitu berisi uraian tinjauan pustaka dan dasar teori yang dibutuhkan serta mendukung penelitian.

2.1 Tinjauan Pustaka

Pada tinjauan pustaka dijelaskan tentang penelitian sebelumnya yang berkaitan dengan penelitian yang akan dilakukan yaitu Rancang Bangun GPS *Back Track* pada Rekaman Rute Pendakian Berbasis Sistem *Embedded*. Tinjauan pustaka yang digunakan sebagai acuan dalam penelitian ini adalah skripsi dengan judul “Aplikasi GPS Berbasis GSM Modem Pada Monitoring BUS” oleh Murie Dwiyaniti (Dwiyaniti, 2011). Pada penelitian lain yang dilakukan oleh Iftitahatul dan Bambang Nurcahyo yang membahas tentang “Uji GPS Tracking Dalam Skala Transportasi Antar Kota”, untuk mengetahui keberadaan kendaraan transportasi. Pada penelitian tersebut dielaskan penggunaan GPS dalam memantau posisi kendaraan dengan menggunakan data kordinat Longitude dan *Latitude*, sebagai parameter untuk mengetahui letak dan posisi kendaraan (Hanifa, 2016). Andronikus (2015) dalam penelitian yang berjudul “Rancang Bangun Aplikasi Penanda Lokasi Peta Digital Gunung Merbabu Berbasis *Mobile GIS* Pada *Smartphone Android*” fungsi GPS dalam kegiatan pendakian masih sangat dibutuhkan. Dengan bantuan GPS yang berfungsi memberikan informasi lokasi dinilai efektif mengurangi resiko pendaki tersesat (Lukito, 2015). Berikut tabel persamaan dan perbedaan dari penelitian yang telah ada dengan penelitian yang akan dilakukan penulis :

Tabel 2.1 Persamaan dan perbandingan

No	Nama	Judul	Persamaan	Perbedaan
1	Murie Dwiyaniti	Aplikasi GPS Berbasis GSM Modem Pada Monitoring Bus	Perangkat keras yang digunakan, Modul GPS Ublox Neo 6M sebagai sensor yang digunakan sebagai pengambil data koordinat <i>Latitude</i> dan <i>Longitude</i> sebagai penentu lokasi.	Penelitian sebelumnya data GPS yang diperoleh dikirim menggunakan modul GSM, penelitian yang akan dilakukan data GPS yang diperoleh akan ditampilkan pada LCD <i>Display</i> kemudian akan disimpan sebagai hasil rekaman.
2	Iftitahatul Hanifa	Uji GPS <i>Tracking</i> Dalam Skala Transportasi Antar Kota	Persamaan terletak pada output dari sistem yang dapat merekam <i>tracking</i> perjalanan dan	Menggunakan dua perangkat GPS. perekaman data GPS menggunakan sistem online dengan <i>output</i>

			digunakan perangkat GPS untuk pengambilan data kordinat <i>Latitude</i> dan <i>Longitude</i> .	tampilan <i>maps</i> . Penelitian yang akan dilakukan data GPS yang terekam akan di proses secara offline dan di tampilkan dalam bentuk grafik yang akan membentuk rute.
3	Andronikus Anggoro Lukito	Rancang Bangun Aplikasi Penanda Lokasi Peta Digital Gunung Merbabu Berbasis Mobile GIS Pada Smartphone Android	Rancang bangun sistem penanda pada rute pendakian dengan menggunakan teknologi GPS	Menggunakan perangkat <i>Mobile android</i> dalam implementasi. Menggunakan komponen GPS, LCD, dan <i>Raspberry pi 2B</i>

Penelitian dilakukan dengan tujuan untuk merancang sebuah alat yang memiliki fungsi merekam rute mendakian berdasarkan data koordinat GPS *latitude* dan *longitude*. Alat ini juga dilengkapi dengan LCD *display* untuk menampilkan informasi rute yang dilalui. Hal ini bertujuan untuk membantu pendaki mengetahui koordinat lokasi dan rute yang dilaluinya. Alat ini dapat memberikan informasi berupa rekaman rute sebagai acuan untuk kembali ke rute awal yang dilaluinya.

Kebutuhan penggunaan teknologi yang semakin berkembang mendorong masyarakat untuk menggunakan perangkat yang bersifat praktis. Salah satu teknologi praktis dan efisien untuk dibawa saat melakukan pendakian yaitu alat bantu navigasi. Untuk merancang sebuah teknologi navigasi yang praktis dan mudah dibawa diperlukan beberapa kebutuhan komponen yang dapat digabung menjadi satu rangkaian alat. Kebutuhan bahan yang akan digunakan adalah *Board Raspberry Pi 2B*, Modul GPS, LCD 5 inch serta komponen penunjang lainnya. Perekaman rute yang akan dibangun adalah sebuah alat bantu navigasi praktis dan dapat dibawa berjalan. GPS akan membaca kordinat rute perjalanan yang kemudian akan direkam oleh *Raspberry Pi 2B* dan ditampilkan pada LCD *display* dalam bentuk garis rute hasil rekaman. Apabila pengguna ingin kembali ke lokasi awal, pengguna hanya perlu mengikuti rute yang telah terekam dengan arah sebaliknya.

2.2 Dasar Teori

Dasar teori adalah teori acuan yang diperlukan dan bersifat relevan mengenai unsur pendukung baik perangkat keras dan perangkat lunak maupun objek

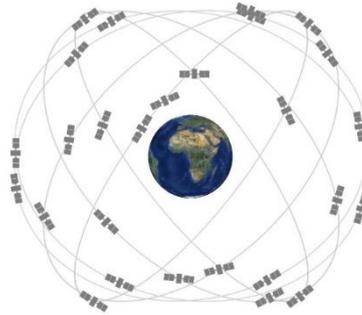
penelitian yang digunakan untuk merancang sebuah sistem yang dapat berjalan dan bekerja pada alat.

2.2.1 Navigasi

Navigasi adalah penentuan posisi dan arah perjalanan baik di peta maupun medan sebenarnya (Silitonga, 2017). Dalam dunia penjelajahan alam navigasi adalah salah satu materi pokok yang wajib dipelajari. Ilmu navigasi Secara umum terbagi menjadi navigasi darat, navigasi laut, dan navigasi udara. Sebagai penganut alam, kemampuan dalam navigasi merupakan hal penting terutama dalam membaca peta dan kompas. Kemampuan dalam navigasi juga digunakan untuk melakukan pertolongan seperti usaha pencarian dan penyelamatan korban kecelakaan atau tersesat di gunung. Perlengkapan yang biasa digunakan dalam navigasi adalah peta dan kompas. Peta dinyatakan sebagai penggambaran dari permukaan bumi dalam bentuk dua dimensi (2D) bidang datar yang terlihat dari atas. Untuk dapat menentukan posisi dengan menggunakan peta, secara manual pendaki harus menggunakan kompas untuk membidik ke dua arah mata angin yang berbeda kemudian digambarkan pada peta untuk dapat mendapatkan titik koordinat lintang dan bujur.

2.2.2 GPS

GPS (*Global Position System*) adalah sistem navigasi berbasis satelit yang didesain untuk menampilkan hasil citra permukaan bumi dari atas yang dapat menyampaikan posisi objek diseluruh permukaan bumi (Mandalamaya, 2015). Secara umum GPS dipergunakan untuk keperluan pelacakan dan navigasi. selain itu GPS juga digunakan untuk keperluan militer, pertanian transportasi dan sipil (Damani, 2015). Sejarah GPS berawal dari peluncuran satelit ke orbit bumi untuk pertama kalinya. Para ilmuwan menemukan bahwa mereka mampu melacak pergerakan satelit dengan mengamati pergeseran sinyal radionya. Kemudian departemen pertahanan Amerika Serikat mulai meluncurkan satelit GPS rahasianya yang diberi Navstar (*Navigation System With Timing and Ranging*). Pada desember 1993 sistem navigasi GPS siap beroperasi secara penuh dengan konstelasi 24 buah satelit di orbit bumi dan sudah dikomersialkan untuk memudahkan navigasi. Dengan adanya GPS semua pihak akan lebih mudah dalam bernavigasi secara digital tanpa harus membuka peta manual dan menghitung koordinat posisi.

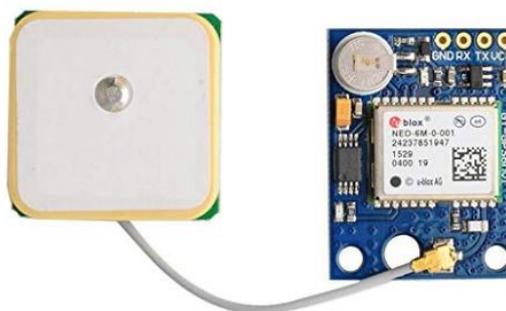


Gambar 2.1 Satelit GPS

Sumber : (Nasa, 2014)

Dalam menentukan posisi GPS memiliki tiga komponen utama yang saling bekerja sama untuk menentukan posisi suatu obyek. Komponen pertama adalah *ground control station* yang berfungsi sebagai stasiun yang terdapat di bumi. Yang kedua komponen luar angkasa dari GPS memakai 24 satelit yang mengorbit pada ketinggian sekitar 20.372 km diatas permukaan bumi. Komponen ketiga adalah *GPS receiver* yang berfungsi sebagai penerima sinyal GPS. *Receiver* inilah yang menerima sinyal dari satelit dan menyediakan posisi dengan komponen antena dan prosesor untuk mengolah data satelit seperti kecepatan dan ketepatan waktu. Untuk menerima data GPS yang dibutuhkan *receiver* GPS minimal harus mendapatkan sinyal dari minimal 3-4 satelit untuk mendapatkan sinyal (Isrianto, 2016).

Pada penelitian ini akan digunakan modul GPS Ublox Neo 6M yang digunakan sebagai komponen *receiver* data koordinat *Longitude* dan *Latitude*. Modul GPS Neo 6 M dilengkapi dengan 4 pin VCC, GND, RX, TX dan antena untuk dapat bekerja pada kontroler utama. Untuk dapat bekerja dengan baik Modul GPS Neo 6M membutuhkan tegangan kerja 0.5 – 3.5 V dan membutuhkan waktu sekitar 27 detik untuk mendapatkan sinyal di ruangan terbuka tanpa penghalang (u-blox).



Gambar 2.2 Fisik GPS Neo 6 M

Sumber : (Amazon, 2018)

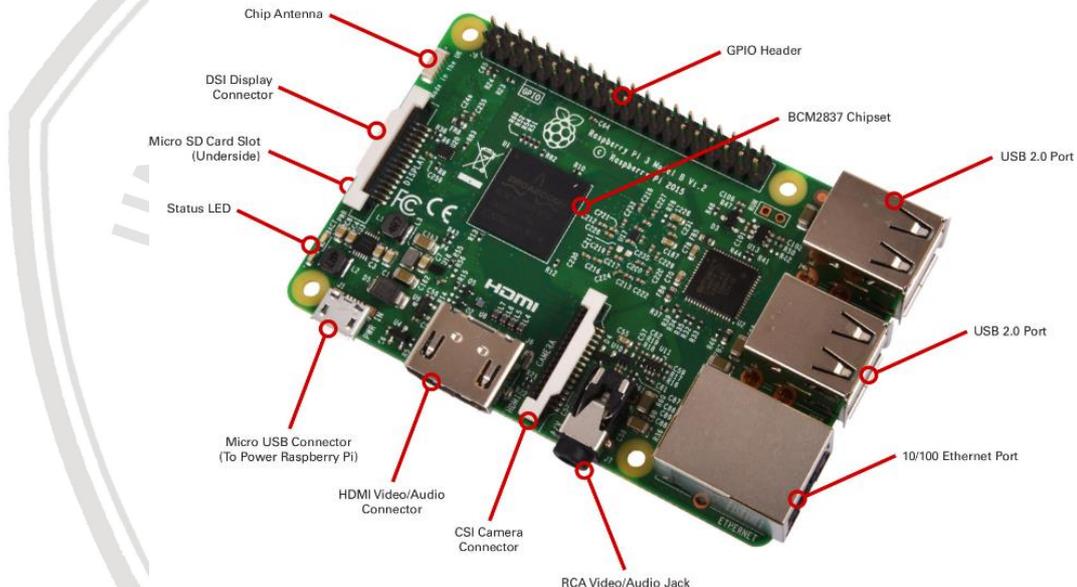
Tabel 2.2 Spesifikasi Ublox Neo 6M Modul

VCC	Tegangan kerja 0.5 - 3.6 Volt
TX	<i>Transmitter</i>

RX	Receiver
GND	Ground
Antena	Receiver sinyal satelit

2.2.3 Raspberry Pi 2B

Raspberry Pi 2B adalah sebuah mini komputer atau *Ingle Board Computer* yang dikembangkan oleh *Raspberry Foundation* yang bertempat di *United Kingdom (UK)* (Utomo, 2016). Berbeda dengan komputer pada umumnya, *Raspberry Pi 2B* memiliki ukuran yang lebih kecil daripada ukuran komputer standar yang dikenal oleh umum. *Raspberry Pi 2B* memiliki spesifikasi RAM berkapasitas 1GB, *Prosesor Quad Core 64-bit* dengan memori penyimpanan menggunakan *SD Card*. *Raspberry Pi 2B* mengalami peningkatan performa hingga 50 % pada seri Pi 3 tipe B jika dibandingkan dengan pendahulunya (Pi2), dan masih memungkinkan dipergunakan dalam membangun perangkat IoT dan *embedded* sistem.



Gambar 2.3 Fisik Raspberry Pi 2B

Sumber : (cnx software, 2016)

Berdasarkan Gambar 2.3 diatas merupakan *raspberry Pi 2B* dengan tampilan 4 x *USB slot*, *Slot SD card memory*, 40 *GPIO pin conector*, *HDMI Conector*, *Port Ethernet*, *CSI Camera Connector*, *Slot micro SD card*, *Micro USB connector*, *Chip antena* dan *status led*. Namun dalam penelitian yang akan dilakukan tidak semua layanan di gunakan. Berikut tabel spesifikasi *Raspberry Pi 2B* (RaspberryPi).

Tabel 2.3 Spesifikasi Rasperry Pi 3

Processor	BCM2837 64-bit Quad Core 1.2 GHz
Tegangan kerja	5 V



Tegangan kerja input	3.3 – 5 V
Tegangan kerja output	3.3 – 5 V
Pin GPIO	40
Wifi	BCM43438
USB port	4
Memory Slot	1 Slot SD Card
Power Micro USB Socket	5 V, 2,5 A
SRAM	1 GB
Clok speed	1,2 GHz

2.2.4 Display LCD

Menurut Muhammad Ali Muzidi, LCD adalah singkatan dari *Liquid Crystal Display* yang dapat menampilkan berbagai hal yang berkaitan dengan mikrokontroler seperti menampilkan teks (Mazidi, 2011). Untuk menghubungkan LCD dengan mikrokontroler dibutuhkan *port* yang saling suport untuk menjadi penghubung antara LCD dengan mikrokontroler. Sebagai contoh spesifikasi LCD yang dapat dihubungkan dengan mikrokontroler digunakan LCD (800X480) inch yang mampu menampilkan status kerja alat yang dikontrol oleh mikrokontroler.



Gambar 2.4 Fisik LCD (800x480) Touch Display for Raspberry Pi 2B

Sumber : (Manual, 2017)

Berdasarkan Gambar 2.4 menampilkan LCD 800 X 480 yang dapat dihubungkan dengan perangkat mikrokontroler untuk dapat bekerja dan mampu digunakan sebagai penampil atau *monitoring* status kerja suatu alat yang dikerjakan oleh mikrokontroler. Adapun deskripsi perangkat keras sebagai berikut:

1. Power soket
2. HDMI interface
3. Backligh power swicth
4. 13 x 2 pin socket
5. Extended 13 x 2 socket signal Pin to Pin

Tabel 2.4 Spesifikasi *Display LCD*

GND	--	Ground
VCC	--	Power supply +5 V DC
HDMI Socket	Input/output	Komunikasi HDMI
Back Light On/Off	Input	Mengatur tampilan pencahayaan latar <i>On/Off</i>
Resolusi	--	800 x 480 (<i>Touch Screen</i>)
Size	--	5 Inch
VCC Input Socket	Input	5 V DC

VCC memberikan supply tegangan kerja 5V, VSS berfungsi sebagai ground, dan Back light *On/Off* berfungsi sebagai pengatur pencahayaan yang dapat di *On* atau *Off* kan dari tampilan. HDMI *Socket* berfungsi sebagai konektor dari perangkat Mikrokontroler (Raspberry Pi 2B) ke LCD untuk menampilkan User Interface. USB soket digunakan sebagai koneksi perangkat tambahan seperti mouse, keyboard dan perangkat lain yang menggunakan USB. 13 x 2 port-pin utama yang terkoneksi langsung dengan pin GPIO *Raspberry Pi 2B*. 13 x 2 pin tambahan yang menggantikan 13 x 2 pin GPIO raspberry yang tertutup port LCD. Perangkat LCD diatas mendukung terhadap Raspberry Pi 2 tipe B dengan resolusi tampilan 800 x 480 Touch Sreen, dan mempunyai ukuran 5 Inch. Pada LCD juga di lengkapi dengan VCC socket Input bilamana pengguna ingin menggunakan sumber tegangan dari perangkat tambahan seperti contoh adaptor DC (*Display, n.d.*).

2.2.5 Bahasa Pemrograman Python

Bahasa pemrograman Python merupakan bahasa yang digunakan dalam pemrograman berorientasi objek yang multiguna, kaya akan *Library* yang dapat digunakan secara gratis dan juga memiliki komunitas diskusi yang cukup besar. Python biasanya digunakan dalam menyusun atau membuat suatu proyek yang berorientasi pada objek atau juga fungsional. Saat ini bahasa python memiliki dua versi yaitu Python 2 dan Python 3. Python 3 dikembangkan secara aktif dengan fitur – fitur baru yang ditambahkan dengan dukungan dari Python 2 yang saat ini juga masih banyak digunakan (Cai, 2005).

Pada saat ini kode pemrograman Python dapat dijalankan pada beberapa *platform* sistem operasi seperti Windows, Linux, Mac OS, Raspbian jessie OS dan yang lainnya. Selain dapat dijalankan dalam beberapa sistem operasi yang umum digunakan oleh orang pada umumnya, Python juga dilengkapi dengan beberapa fitur yang memudahkan pengguna dalam menyusun kode program sesuai dengan kebutuhan pengguna seperti *library* modul yang siap pakai untuk berbagai keperluan, memiliki *layout* yang memudahkan pengguna dalam melakukan pengecekan dan pembacaan kode program yang dibuat.

Dalam penelitian yang akan dilakukan, digunakan *Raspberry pi 2B* dengan sistem operasi Raspbian Jessie yang memiliki tampilan seperti Linux yang memungkinkan untuk menyusun dan menjalankan kode program dengan bahasa pemrograman Python sesuai dengan kebutuhan dalam penelitian dengan memanfaatkan *library* yang telah tersedia dalam Python.

2.2.6 Library Modul

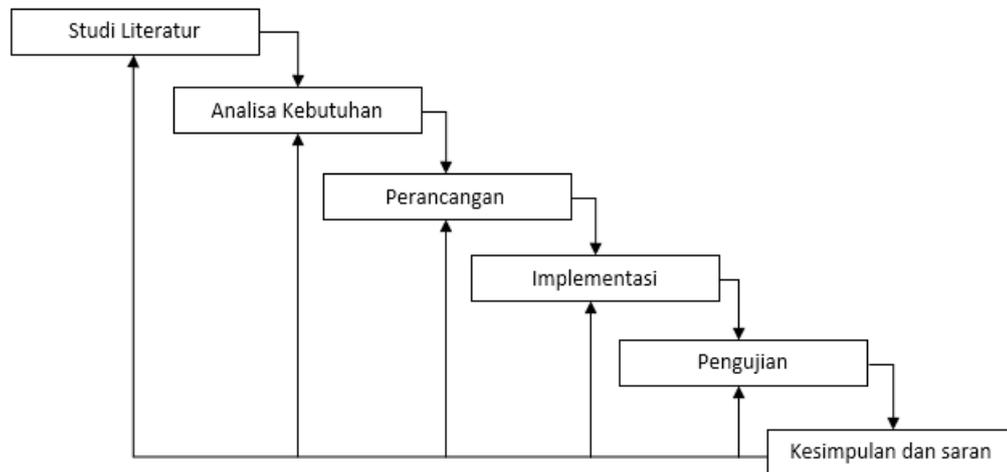
Untuk membuat kode program pada bahasa Python digunakan *library* agar program dapat berjalan pada alat yang digunakan, yaitu Raspberry Pi 2B. *Library* ini berisi tipe data yang dianggap sebagai bagian dari inti suatu bahasa seperti angka dan daftar. Dalam implementasi *library* pada Python biasanya digunakan sintak *import* untuk memanggil *library* yang digunakan pada program untuk dapat menjalankan kode program. Contoh *library* yang digunakan untuk membuat nilai *random* dapat digunakan *import random*. Namun tidak semua *library* pada Python dalam penggunaannya harus menggunakan *import* untuk dapat digunakan karena sudah *built-in* didalam bahasa pemrograman (Foundation, 2018). Pada penelitian yang akan dilakukan, digunakan beberapa *library* untuk membuat kode program yang akan digunakan untuk menjalankan beberapa modul sensor pada Raspberry Pi 2B. *Library tkinter* digunakan untuk membuat tampilan *User interface*, *matplotlib* digunakan untuk membuat kode program pengolah hasil akuisisi data dari sensor untuk dapat ditampilkan dalam bentuk grafik.

2.2.7 Kalman Filter

Kalman filter merupakan persamaan matematis dari sebuah proses filter data linear diskrit untuk menghasilkan perhitungan rata – rata rekursif yang lebih efisien dalam proses *filtering* data dengan tujuan untuk mengurangi terjadinya error. Kalman filter digunakan sebagai estimator rata – rata error dengan membandingkan hasil pengukuran dengan membutuhkan data waktu sebelumnya dan data pengukuran yang sedang berlangsung (Wicaksono, 2009). Pada tahun 2003, kalmanfilter digunakan oleh D. Loebis, R. Sutton untuk melakukan penelitian penerapan kalman filter dalam bidang navigasi cerdas yang mengintegrasikan GPS dan beberapa sensor untuk membuat sistem navigasi cerdas pada perangkat otonom bawah air (Loebis, 2004). Proses yang terjadi kalman filter adalah kalman akan melakukan filter data dengan menggunakan prinsip umpan balik dengan mengestimasi sebuah proses data yang kemudian akan mendapatkan nilai untuk umpan balik berupa hasil pengukuran yang masih bercampur data nois dari sebuah sistem.

BAB 3 METODOLOGI

Untuk mencapai tujuan dalam pelaksanaan penelitian, diperlukan perencanaan kegiatan secara urut mulai dari awal hingga akhir didalam diagram alir berikut :



Gambar 3.1 Diagram Alir Metodologi Penelitian

Berdasarkan Gambar 3.1 diuraikan langkah – langkah yang akan dilakukan secara bertahap dengan skemati diagram *waterfall*. Diawali dengan Studi literatur untuk mengumpulkan informasi dan dasar teori yang mendukung penelitian. Selanjutnya melakukan analisis kebutuhan untuk menentukan apa yang dibutuhkan sesuai dengan spesifikasi sistem yang akan di buat. Kemudian dilakukan perancangan sistem sistem yang akan dibuat sesuai dengan yang sudah ditentukan. Setelah perancangan selesai dilakukan implementasi sistem yang dirancang. Setelah implementasi sesuai dilanjutkan dengan pengujian sistem yang dibuat. kemudian dilanjutkan dengan menarik kesimpulan dan saran berdasarkan hasil proses penelitian. Apabila terdapat ketidaksesuaian dalam langka yang dilakukan dapat meninjau kembali langkah sebelumnya yang telah dilaksanakan.

3.1 Studi Literatur

Studi literatur menjelaskan mengenai dasar – dasar teori yang digunakan sebagai pendukung dalam mengimplementasikan sistem mulai dari perangkat keras, perangkat lunak dan kebutuhan fungsional maupun non fungsional yang sesuai dengan spesifikasi Rancang Bangun GPS Back Track Pada Rekaman Rute pendakian Berbasisi Sistem Embedded. Studi literatur untuk penelitian ini sudah dijelaskan pada bab 2, dan dengan adanya studi literatur diharapkan mempermudah perancangan sistem yang akan dibuat.

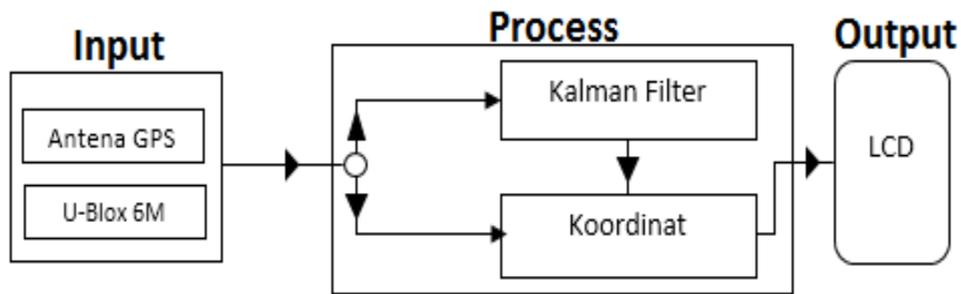
3.2 Analisis Kebutuhan Sistem

Analisis kebutuhan diperlukan untuk menentukan perangkat pendukung yang akan digunakan dalam merancang pembangunan sistem baik perangkat keras, perangkat lunak, kebutuhan fungsional maupun nonfungsional dengan spesifikasi sesuai dengan kebutuhan yang akan digunakan untuk perancangan dan pembangunan sistem. Perangkat yang digunakan sebagai berikut :

1. Kebutuhan perangkat keras
 - a. Raspberry Pi 2B , sebagai kontrol utama sistem.
 - b. GPS modul NEO-6M, sebagai sensor yang membaca lokasi.
 - c. LCD *display*, untuk menampilkan informasi rute.
 - d. Laptop, sebagai perangkat yang mendukung proses perancangan.
 - e. Toolkit, sebagai penunjang dan alat bantu dalam perancangan.
 - f. Power bank, sebagai sumber daya pada alat yang dibangun.
2. Kebutuhan perangkat lunak
 - a. RaspberryOS, perangkat lunak yang digunakan sebagai pemrograman sistem yang akan di bangun pada Raspberry Pi 2B..
 - b. *Geany text editor* perangkat lunak yang akan digunakan untuk melakukan *editing* dan analisis kode program.
 - c. Driver Display LCD digunakan untuk dapat menjalankan LCD pada Raspberry pi.
 - d. MobaXterm sebagai perangkat lunak yang digunakan untuk melakukan remote pada Raspberry pi dengan laptop.

3.3 Perancangan Sistem

Pada perancangan sistem, akan dijelaskan mengenai alur perancangan berdasarkan perangkat keras dan perangkat lunak yang digunakan untuk membangun sebuah sistem yang dapat bekerja pada alat. Perancangan perangkat keras dilakukan secara sistematis sehingga, sistem yang akan dibangun dapat bekerja sesuai dengan fungsi yang diharapkan. Begitu pula dengan perancangan perangkat lunak yang akan dibuat diharapkan dapat menjalankan alat dengan menggunakan pemrograman Python. Perancangan sistem dilakukan supaya alat yang dibuat berjalan sesuai tujuan penelitian. Gambaran perancangan sistem pada alat dapat dilihat dalam diagram blok berikut.



Gambar 3.2 Diagram Blok Sistem

Alat dirancang menggunakan modul GPS U-blox Neo 6 M sebagai sensor yang nilai keluarannya menjadi masukan Raspberry pi untuk selanjutnya diproses menggunakan program yang dibuat menggunakan bahasa pemrograman Python. Data masukan yang diproses oleh Raspberry pi adalah nilai koordinat *latitude* dan *longitude* yang diolah menggunakan kalman filter serta tanpa menggunakan kalman filter kemudian disimpan sebagai hasil rekaman. kemudian keluaran dari sistem akan ditampilkan pada LCD dalam bentuk rute pendakian. Perancangan sistem di atas menggambarkan alur dan proses sistem yang akan berjalan pada alat.

3.4 Implementasi Sistem

Apabila tahap perancangan telah dilakukan maka dilanjutkan ke tahap implementasi. Pada tahap implementasi dilakukan untuk menerapkan hasil perancangan sehingga sistem pada alat dapat bekerja sesuai dengan fungsi yang diharapkan :

1. Perangkat keras yang digunakan dapat berjalan secara keseluruhan sesuai fungsinya masing – masing berdasarkan spesifikasi *datasheet* perangkat keras tersebut.
2. Sistem yang dirancang mampu merekam rute yang dilewati dan ditampilkan pada LCD display.

3.5 Pengujian dan Analisis

Pengujian dilakukan untuk mengetahui kesesuaian dan performa sistem yang dirancang. Kesesuaian merupakan kemampuan sistem dalam menghasilkan output sesuai dengan input dan proses yang diberikan oleh sistem. Performa adalah kemampuan sistem dalam mengolah inputan sehingga menghasilkan output dengan baik sesuai yang diharapkan dengan waktu yang singkat.

3.6 Kesimpulan dan Saran

Kesimpulan diambil berdasarkan hasil pengujian dan analisis sistem yang telah dilakukan sebelumnya. Kesimpulan yang diambil harus mampu menjawab permasalahan yang telah dijelaskan pada bab 1, serta menyimpulkan hasil keseluruhan dari perancangan sistem yang digunakan secara layak dan baik. Selain

kesimpulan pada tahap ini, berisi saran untuk pembaca dengan harapan apabila pembaca ingin mengembangkan sistem yang telah dirancang dapat menggunakan saran sebagai acuan untuk penelitian selanjutnya



BAB 4 REKAYASA KEBUTUHAN

Pada bab ini dijelaskan mengenai syarat yang harus dipenuhi mulai dari tahap perancangan hingga tahap implementasi sistem. Perancangan dan implementasi bertujuan agar system dapat bekerja dan berjalan sesuai dengan perancangan yang dibuat.

4.1 Gambaran Umum Sistem

GPS back track ini ditujukan kepada pendaki awam yang tidak memiliki kemampuan ilmu navigasi darat yang mumpuni, sehingga dalam melakukan kegiatan pendakian di daerah yang belum dikenali dapat merekam jalur yang dilewati. Apabila pendaki mengalami kesalahan dalam melintasi jalur dan tersesat dapat kembali ke jalur yang telah dilewati dengan rekaman yang dibuat. Pada alat GPS ini dilengkapi dengan pilihan tombol, *track*, *Back Track*, *Stop*. Alat GPS ini dirancang tanpa menggunakan jaringan komunikasi dari penyedia layanan telekomunikasi. Dengan pertimbangan alat yang dirancang akan digunakan pada lingkungan dan area yang sulit dapat mengakses jaringan komunikasi dari penyedia layanan komunikasi. Untuk dapat menjalankan perangkat, pengguna harus menjalankan kode program yang dibuat. Dalam kode program dirancang tampilan yang memudahkan pengguna dalam mengoperasikan alat yang dibuat dengan tampilan grafik untuk memantau rute pendakian yang sedang atau telah dilewati. Alat dilengkapi LCD *display touch screen*, sehingga pengguna dapat memilih menu yang disediakan dengan cara menyentunya pada LCD. Untuk melakukan rekaman rute, pengguna dapat menyentuh menu *Track* pada tampilan, apabila pengguna ingin merekam rute kembali dengan mengikuti rute yang direkam sebelumnya, pengguna dapat menyentuh menu *trackback*. Menu *Track* dan *Trackback* didesain memiliki warna berbeda untuk membedakan hasil rekaman rute. Apabila pengguna ingin mengakiri proses perekaman, dapat menyentuh menu *Stop* pada LCD. Maka proses perekaman akan berhenti dan hasil perekaman dapat dilihat pada LCD dalam tampilan grafik yang membentuk rute selama pendakian. Desain sistem pada alat yang dirancang diharapkan sesuai dengan kebutuhan pengguna dan mampu membantu pengguna dalam kegiatan pendakian.

4.2 Identifikasi Pengguna

Identifikasi pengguna yang dapat menggunakan alat ini adalah seorang awam penggiat alam dalam lingkup pendaki gunung. Identifikasi pengguna bertujuan untuk menilai pengujian dari alat berdasarkan pengguna dari alat yang telah dilakukan.

4.3 Kebutuhan Fungsional

Kebutuhan fungsional *GPS back track* adalah kebutuhan yang harus terpenuhi dan bagaimana *GPS back track* dapat merespon tindakan yang dilakukan oleh pengguna. Berikut merupakan fungsi dari *GPS back track* :

1. Pengguna dapat melakukan perekaman rute yang dilalui saat pendakian.

Dengan adanya alat ini pengguna dapat merekam rute pendakian yang dilalui dengan cara memanfaatkan menu *Track* yang terdapat pada tampilan LCD. Dengan demikian rute pendakian akan terekam oleh sistem.

2. Pengguna dapat mengoperasikan fungsi pada tampilan dengan adanya *display LCD Touch Screen*.

Dengan menggunakan *display LCD Touch Screen 800 x 480* pengguna dapat menjalankan menu pada tampilan sistem yang dibuat sehingga tidak memerlukan Mouse untuk menjalankan fungsi *button click*.

3. Pengguna dapat mengetahui rute yang terekam dengan adanya LCD.

Dengan adanya LCD 5inc pada alat pengguna dapat mengetahui hasil rekaman sistem yang ditampilkan pada layar tampilan dalam bentuk informasi grafik rute dan informasi koordinat *Latitude* dan *Longitude*.

4. Pengguna dapat menghentikan perekaman dengan adanya tombol stop.

Pada tampilan *User Interface* dibuat fungsi *pus button stop* sehingga pengguna dapat menghentikan perekaman rute setelah tiba di lokasi pendakian untuk menghentikan perekaman guna menghindari alat melakukan perekaman terus menerus.

5. Pengguna dapat menyimpan hasil rekaman rute pendakian yang dilakukan.

Dalam perekaman rute selama pendakian, data hasil perekaman secara akan tersimpan pada folder yang terdapat pada Raspberry Pi 2B.

4.4 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang akan digunakan alat GPS, sebagai berikut :

1. Raspberry Pi 2B

Digunakan sebagai pemroses data dari sensor GPS yang dikirim melalui komunikasi GPIO Rx dan Tx.

2. Modul GPS Neo 6M

Digunakan sebagai penerima data GPS yang dapat mendeteksi lokasi dengan menangkap sinyal navigasi dalam bentuk lintang (*Latitude*) dan bujur (*Longitude*).

3. LCD 800 x 480

LCD HDMI 5inch 800 x 480 digunakan sebagai *User Interface* yang dapat menampilkan dan memudahkan pengguna melihat informasi rute pendakian yang sedang dilalui dan direkam.

4. Laptop

Laptop dibutuhkan untuk membantu mempermudah pengerjaan rancang bangun alat yang di buat.

5. Toolkit (jumper,solder,timah, dll)

Toolkit seperti kabel jumper, timah, digunakan sebagai penghubung dari sensor ke raspberry. Dan kebutuhan lain yang menunjang dalam pembangunan alat.

6. Power bank

Power bank dibutuhkan sebagai sumber tegangan untuk menjalankan alat yang di bangun.

4.5 Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan GPS *Back track* adalah Raspbian jessie OS, untuk menjalankan Raspberry Pi 2B. Dalam pembuatan program pada Raspberry Pi 2B digunakan perangkat lunak yang dapat digunakan untuk membuat bahasa program dalam format Python (.py) . *Geany text editor* digunakan untuk merancang, *editing*, analisis kode program pada sistem dan, mendesain display tampilan yang dapat memudahkan pengguna dalam pengoprasian alat.

1. Raspbian Jessie OS

RaspbianOs digunakan sebagai sitem oprasi untuk dapat menjalankan *Raspberry Pi* sebagaimana fungsi dari Raspberry Pi 2B sebagai mini PC sehingga dapat digunakan untuk memproses data dan menjalankan perangkat modul sensor yang digunakan.

2. Geany Text Editor

Geany text editor digunakan untuk melakukan *editing* dan analisis kode program yang akan diimplementasikan pada sistem.

3. Driver Display LCD 5inc 800x480

Driver display Lcd digunakan pada Raspberry Pi 2B untuk dapat menjalankan fungsi *Touch screen* pada LCD dan mengaktifkan pengaturan tampilan yang tersedia pada LCD.

4. MobaXterm

MobaXterm digunakan untuk mengoprasikan Raspberry Pi 2B dengan menggunakan SSH yang terhubung dengan *wifi*. Sehingga proses perancangan dan implementasi kode program dapat dilakukan melalui laptop.

4.6 Batasan Desain Sistem

Sistem yang didesain merupakan *prototype*, dan pengujian dilakukan oleh penulis. Dalam sistem ini, penulis merupakan pengguna tunggal yang melihat dan mengamati hasil keluaran sistem yang ditampilkan melalui LCD dikarenakan

sistem yang dibuat masih dalam bentuk *prototype*. Untuk memberi ruang lingkup yang aman dalam implementasi sistem maka dibuat batasan dari desain sistem sebagai berikut :

1. Raspberry Pi 2B digunakan untuk memproses data dari sensor untuk dapat di tampilkan pada LCD.
2. Data masukan yang akan diproses sebagai keluaran sistem dari sensor GPS adalah *Longitude* dan *Latitude*.
3. Keluaran dari sistem yang ditampilkan pada LCD sebagai keluaran sistem adalah tampilan rute dalam bentuk garis.
4. Sistem *gps back track* pada rekaman rute pendakian merupakan *prototype* yang belum diterapkan pada kondisi yang sebenarnya.



BAB 5 PERANCANGAN DAN IMPLEMENTASI

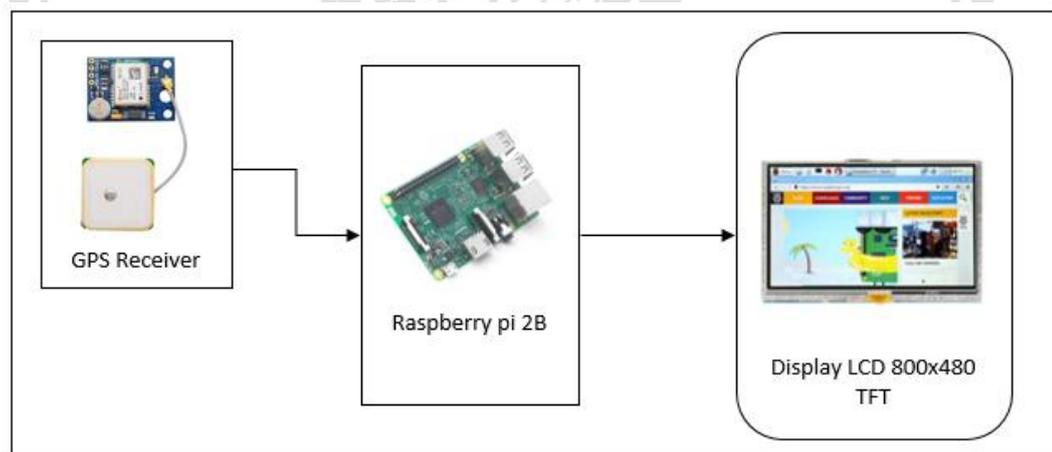
Bab ini membahas perancangan dan implementasi yang dilakukan pada penelitian ini. Perancangan membahas tentang persiapan dan instalasi sehingga sistem yang dirancang siap diimplementasikan. Implementasi membahas mengenai penerapan dari sistem sesuai dengan perancangan yang telah dilakukan sebelumnya.

5.1 Perancangan

Pada bagian ini berisi perancangan secara perangkat keras dan perangkat lunak yang nantinya akan diterapkan pada penelitian “Rancang Bangun GPS *Track Back* pada Rekaman Rute Pendakian Menggunakan Sistem *Embedded*”. Perancangan dilakukan meliputi, perancangan perangkat keras dan perancangan perangkat lunak.

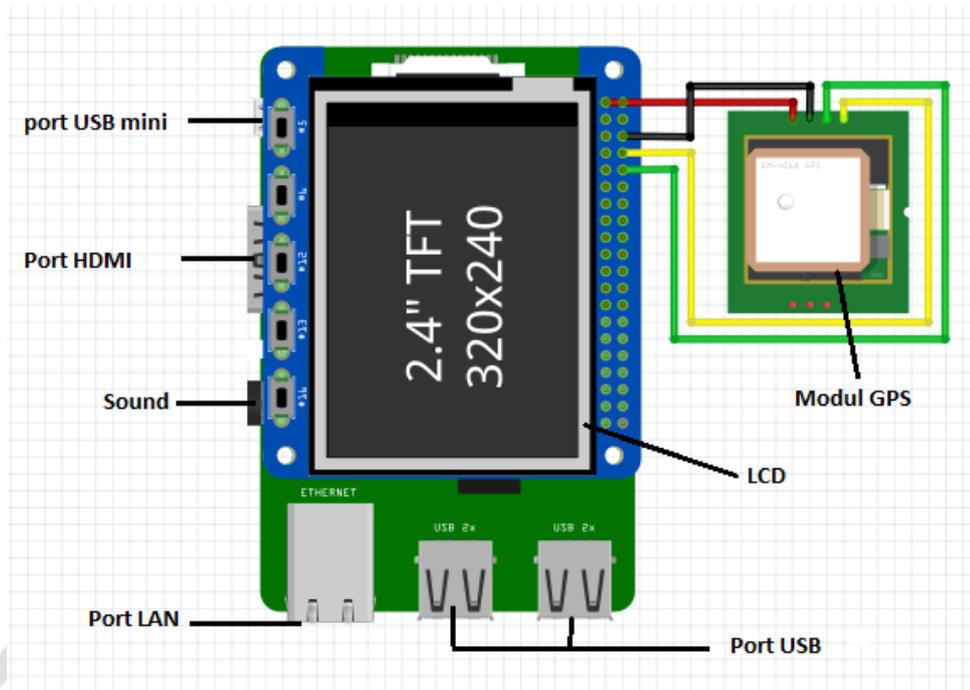
5.1.1 Perancangan Perangkat Keras

Perangkat keras yang digunakan pada sistem ini adalah *Raspberry Pi 2B* dan berfungsi sebagai pengolah data masukan dari sensor. Perangkat yang digunakan sebagai input adalah modul GPS U-blox Neo-6m. Modul GPS U-blox Neo-6m digunakan untuk mengambil data GPS *Latitude* dan *Longitude* yang digunakan untuk menentukan titik kordinat lokasi. Kemudian untuk menampilkan hasil pengolahan data digunakan LCD *display* 800 x 480 TFT. Perancangan perangkat keras digambarkan dalam diagram blok sebagai berikut :



Gambar 5.1 Diagram Blok Perancangan perangkat keras

Diagram blok perancangan sistem menjadi acuan dalam perancangan sistem yang dibuat. Komponen penunjang perancangan sistem berupa modul GPS, Raspberry Pi 2B, *Display LCD 800 X 480 TFT* dirangkai menjadi satu sehingga dapat saling terhubung dan bekerja sesuai dengan fungsinya dan digambarkan sebagai berikut :



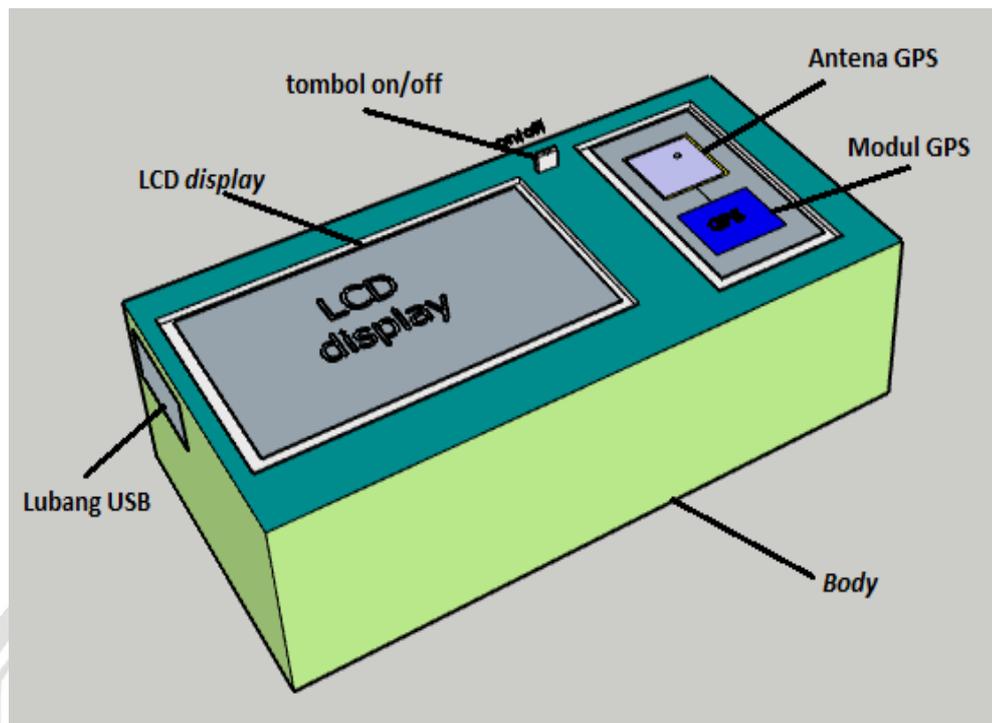
Gambar 5.2 Perancangan perangkat keras

Untuk dapat bekerja sesuai dengan fungsi, seluruh komponen perangkat keras harus saling terhubung dengan spesifikasi port pin dari masing-masing komponen. Seluruh modul yang digunakan dihubungkan dengan Raspberry Pi 2B sebagai pusat pengolahan data utama pada alat. Modul Ublox GPS Neo 6M dan Display LCD dihubungkan pada pin GPIO Raspberry Pi 2B menggunakan kabel sesuai dengan spesifikasi pin komunikasi masing – masing modul. Berikut adalah tabel komunikasi antar pin :

Tabel 5.1 Komunikasi pin GPIO

No	Raspberry	Display LCD	Ublox GPS Neo 6M
1	VCC	VCC	VCC
2	GND	GND	GND
3	RX	-	TX
4	TX	-	RX
5	SDA	-	-
6	SCL	SCL	-
7	HDMI	HDMI	-
8	MISO	MO	-
9	MOSI	MI	-
10	IRQ	IRQ	-

11	CS	CS	-
----	----	----	---

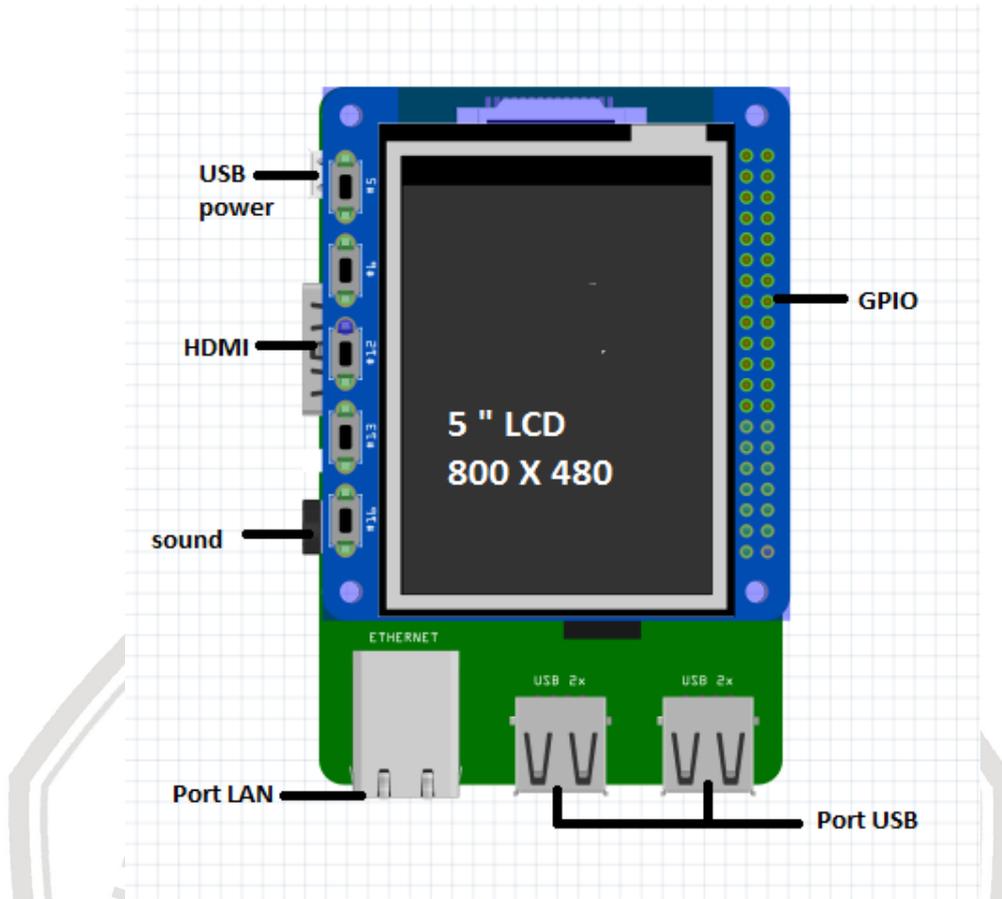


Gambar 5.3 Desain Kemasan Sistem

Untuk memudahkan sistem yang dirancang agar lebih mudah dibawa dan digunakan maka dibuat desain kemasan dari sistem yang digunakan sebagai wadah dari rangkaian sistem tanpa mempengaruhi kinerja sistem.

a) Perancangan Rangkaian *Display* LCD

Pada perancangan rangkaian *display* digunakan *display* LCD 800 X 480 *touch screen*. LCD akan dirangkai dengan Raspberry Pi 2B dengan menggunakan komunikasi HDMI. Pada *display* LCD yang akan digunakan terdapat beberapa pin komunikasi diantaranya 13 x 2 pin GPIO, 1 buah konektor HDMI dan, 1 buah konektor USB power. Desain perancangan *display* LCD dengan Raspberry Pi 2B seperti gambar 5.4 berikut.

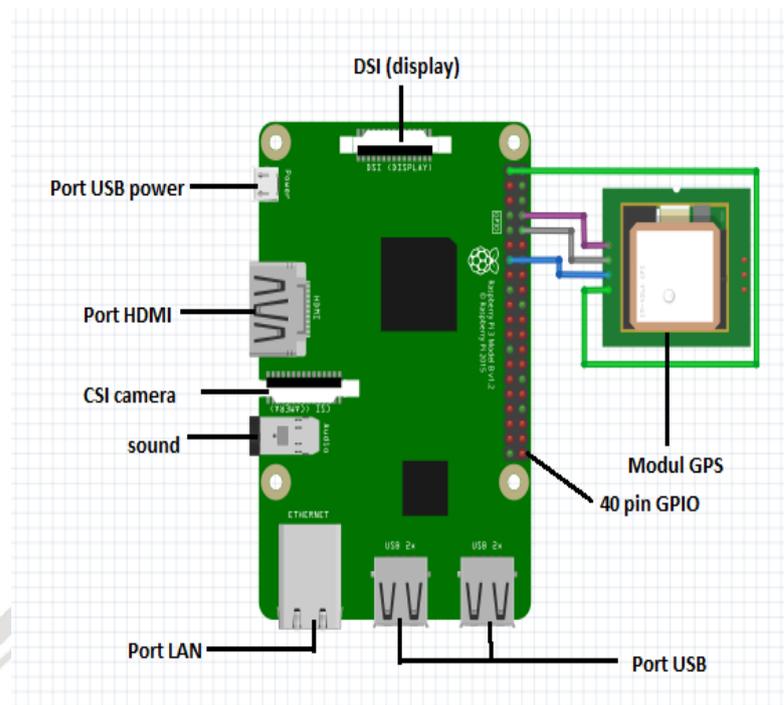


Gambar 5.4 Desain perancangan *display* LCD

Port HDMI pada LCD akan dihubungkan dengan port HDMI pada Raspberry Pi 2B dan pin GPIO pada LCD akan dihubungkan dengan pin GPIO pada Raspberry Pi 2B. fasilitas *Touch Screen* pada LCD dapat berfungsi dengan menghubungkan pin GPIO TPInterrupt(IRQ) 21, TPChipsellect(CS) 26, MI(MOSI) 19, MO(MISO) 21, SCLK 23, VCC 5V 2, GND 25. Dengan desain perancangan pada *display* LCD diharapkan LCD yang digunakan dapat berfungsi sesuai dengan harapan yang diinginkan.

b) Perancangan Rangkaian Modul GPS

Pada perancangan rangkaian modul GPS akan digunakan modul GPS *Ublox Neo 6 M* sebagai sensor pengolah data GPS. Modul GPS *Ublox Neo 6 M* memiliki 4 pin komunikasi UART yang dapat diimplementasikan dengan Raspberry Pi 2B. Modul GPS yang akan digunakan dihubungkan dengan Raspberry Pi 2B menggunakan pin VCC, RX, TX dan, GND. Gambar 5.5 merupakan desain perancangan rangkaian modul GPS dengan Raspberry Pi 2B sebagai berikut.



Gambar 5.5 Perancangan rangkaian modul GPS

Desain perancangan diatas dijelaskan bagaimana modul GPS yang akan digunakan dapat berfungsi dengan Raspberry Pi 2B. Modul GPS Ublox Neo 6 M bekerja pada tegan 3.3 Volt. Kabel berwarna hijau sebagai VCC akan dihubungkan dengan pin GPIO 3.3 Volt pada Raspberry Pi 2B. Sedangkan, kabel warna ungu merupakan RX akan dihubungkan ke pin GPIO TX dan kabel warna abu – abu sebagai TX akan dihubungkan dengan pin GPIO RX pada Raspberry Pi 2B. Kemudian kabel berwarna biru sebagai GND akan dihubungkan dengan pin GPIO GND pada Raspberry Pi 2B. Dengan desain perancangan rangkaian modul GPS diharapkan modul GPS yang digunakan dapat bekerja dengan Raspberry Pi 2B.

Tabel 5.2 Pin komunikasi modul GPS

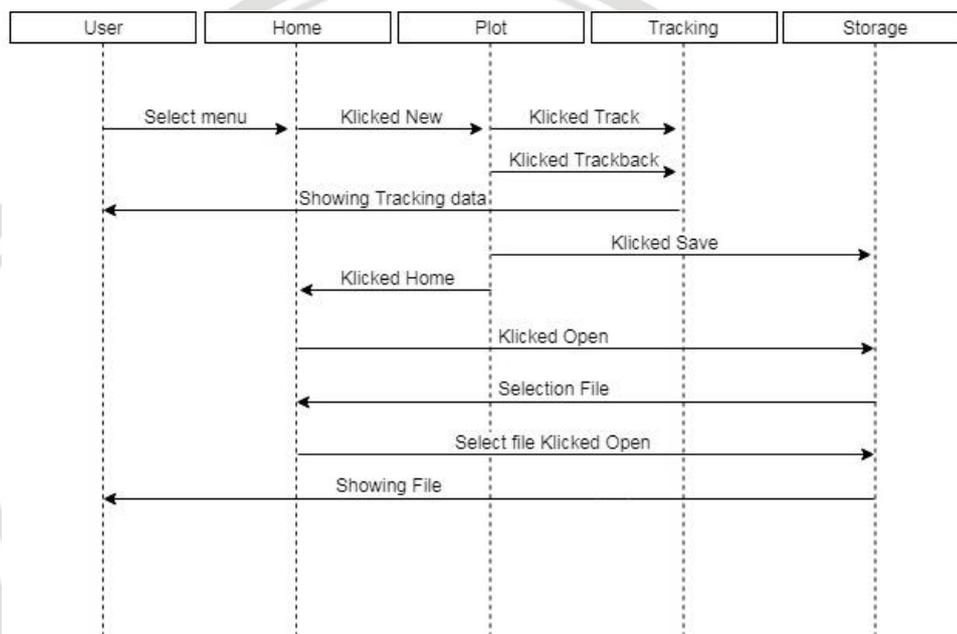
No	Pin GPIO	Pin Modul GPS
1	VCC 3.3 V	VCC
2	GND	GND
3	TX	RX
4	RX	TX

5.1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak akan membahas perancangan sistem yang dibangun untuk menjalankan perekaman rute pendakian pada alat dengan menggunakan perangkat lunak yang telah dibahas pada bab sebelumnya. Perancangan perangkat lunak bertujuan merancang proses yang akan dikerjakan oleh sistem guna mendapatkan input dari perangkat keras modul GPS yang digunakan untuk diproses oleh Raspberry pi 2B berupa data rekaman rute dan

mendapatkan output yang dapat ditampilkan pada LCD *display* sesuai dengan yang diharapkan. Perancangan yang dilakukan akan dijadikan acuan dalam pembangunan sistem dengan menggunakan bahasa pemrograman Python dan memanfaatkan beberapa *library* penunjang yang digunakan untuk menampilkan data hasil rekaman.

Pada rancang bangun GPS *back track* pada rekaman rute pendakian menggunakan sistem *embedded* diperlukan sebuah sistem yang dapat mengolah data GPS berupa nilai koordinat posisi yang dirubah dalam bentuk garis rute pendakian dan menampilkannya pada LCD *display*. Dalam perancangan sistem ini akan dirancang sebuah *user interface* yang berjalan pada alat beserta sistem pemrosesan alat yang digambarkan dalam sebuah diagram *activity*. Pada gambar 5.6 digambarkan alur aktifitas sistem yang akan berjalan.

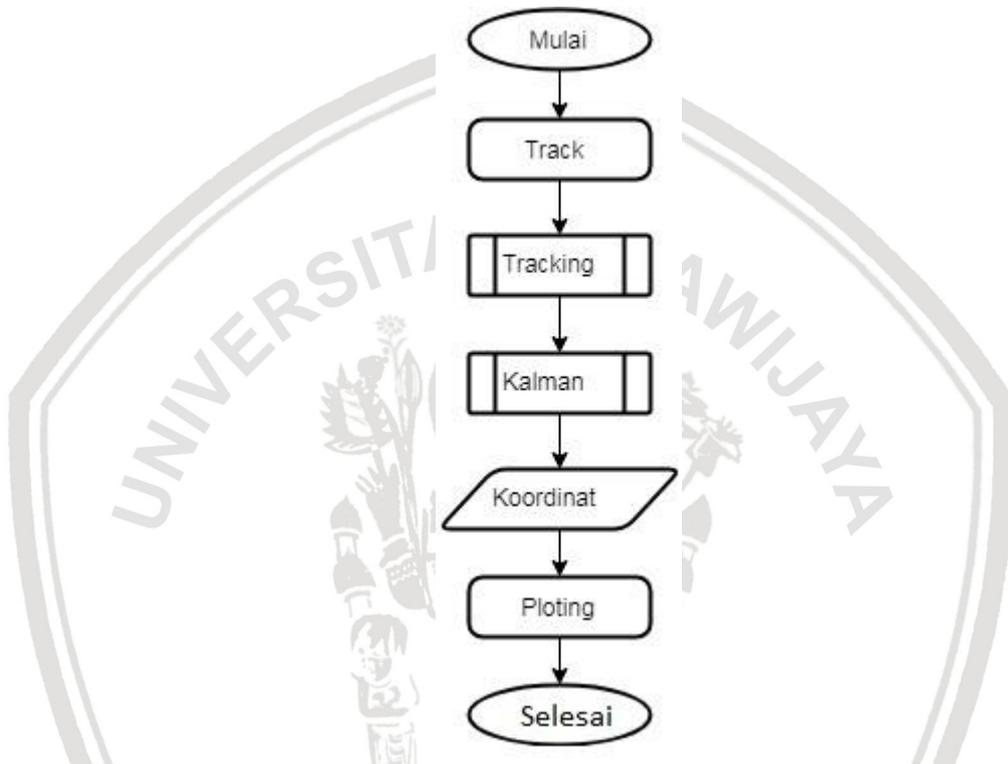


Gambar 5.6 Diagram Aktivitas Sistem

Pada diagram *activity*, sistem yang akan berjalan pada alat dirancang memiliki tampilan *user interface* yang mempunyai dua frame. Pada frame pertama akan digunakan sebagai halaman *Home* yang menampilkan tombol *NEW*, *OPEN* dan *EXIT*. Kemudian pada frame kedua sebagai halaman *graph* yang menampilkan oprasi utama pada alat dalam melakukan perekaman rute yaitu display *Ploting*, tombol *Track*, *Trackback*, *Save*, *Home* dan *Exit*. Alat yang dirancang, menggunakan LCD *display* TFT 800x480 yang dibekali dengan teknologi layar sentuh sehingga dapat mempermudah pengguna dalam melakukan proses perekaman dengan sistem yang berjalan pada alat. Alur dari proses perekaman diawali pada sistem *Standby* dapat memilih menu pada halaman *HOME*, *click New* untuk masuk pada halaman *graph*. Pada halaman *graph* terdapat beberapa pilihan menu seperti yang telah disebutkan diatas. *Click Track* atau *Trackback* untuk memberikan perintah sensor GPS melakukan *Tracking* data koordinat, sensor GPS akan terus melakukan pengambilan data koordinat hingga pengguna menghentikannya dengan cara *click Stop*. Data koordinat yang peroleh dari sensor akan diproses dengan kalman filter

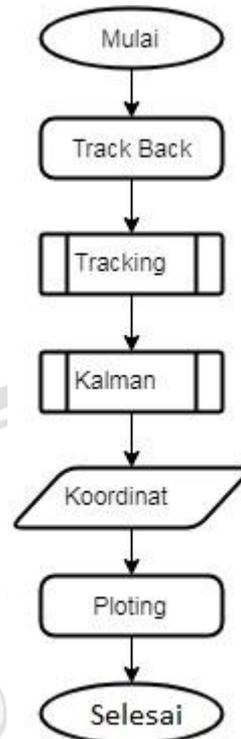
untuk dilakukan penyaringan data koordinat. Hasil dari proses penyaringan berupa data koordinat update yang kemudian disimpan dalam folder file *temporary* dan kemudian akan di*Ploting* oleh sistem menjadi tampilan garis rute. Data koordinat selain di*Ploting* menjadi garis koordinat, data juga akan disimpan kedalam folder file penyimpanan jika pengguna dengan *click Save*. Kemudian alur untuk proses membuka kembali file data hasil perekaman, pengguna dapat menggunakan fungsi *button click OPEN*. Kemudian pengguna dapat memilih file hasil rekaman yang telah disimpan untuk dibuka kembali dan ditampilkan pada LCD *display*.

a) Perancangan *Ploting Rute Dengan Fungsi Track*



Gambar 5.7 *Flowchart Ploting Rute Track*

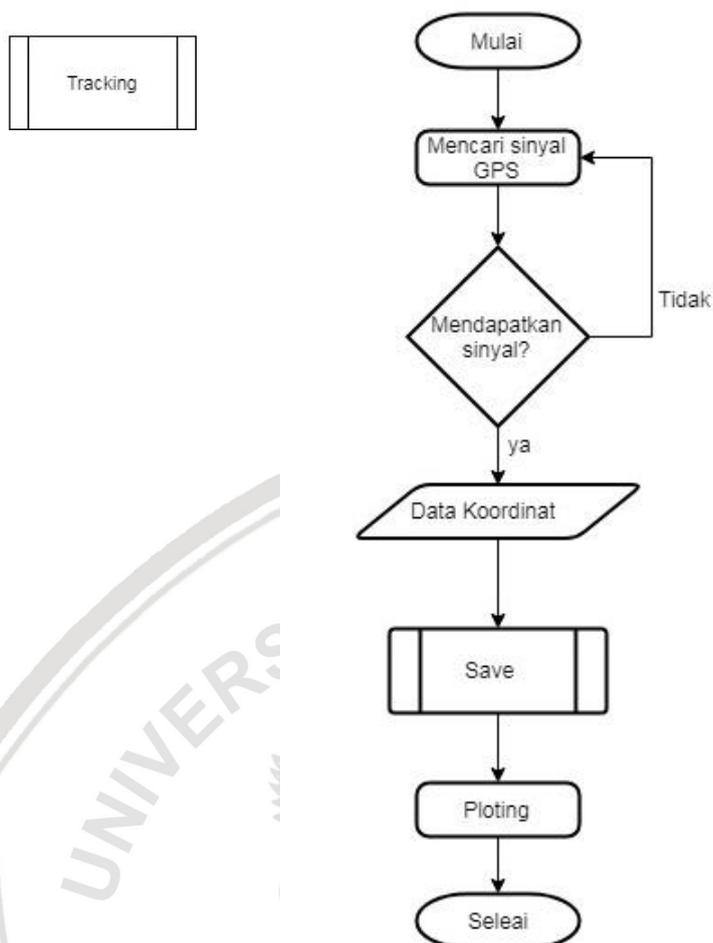
Gambar 5.7 merupakan gambaran proses *Ploting* rute dengan menggunakan fungsi *track*. Pada saat melakukan perekaman rute yang akan dilalui pengguna akan merekam rute perjalanan awal dengan menggunakan fungsi *track*. Fungsi *track* digunakan untuk melakukan *Ploting* data koordinat dalam bentuk garis rute yang menjadi acuan awal perjalanan. Data yang diperoleh dari sensor GPS diolah dengan menggunakan kalman filter untuk mendapatkan nilai koordinat yang telah diupdate oleh kalman filter. Kemudian data disimpan dalam file temporer yang kemudian dibaca oleh sistem untuk di*Ploting* menjadi garis rute dengan warna biru.

b) Perancangan *Ploting* Rute Dengan Fungsi *Trackback***Gambar 5.8 Flowchart *Ploting* Rute *Trackback***

Gambar 5.8 menggambarkan perancangan *Ploting* rute dengan menggunakan fungsi *trackback* yang dapat melakukan *Ploting* data koordinat untuk menuntun pengguna kembali ke rute awal ketika rute yang dilewati menyesatkan pengguna atau menemui jalan buntu. Sehingga pada sistem dirancang sebuah fungsi yang dapat melakukan tracking data GPS dan kemudian diolah menjadi garis rute yang ditampilkan pada LCD *display*. Serupa dengan proses *Ploting* rute awal, pada sistem dirancang fungsi *trackback* yang digunakan untuk mengolah data GPS menjadi garis rute perjalanan dengan warna garis orange.

c) Sub-Program *Tracking* Data GPS

Tracking Data GPS adalah perancangan yang digunakan untuk proses pengambilan data GPS dari satelit dalam menentukan koordinat *Latitude*, *Longitude*. Proses pengambilan data dilakukan oleh modul GPS Ublox Neo 6M dengan menerima sinyal dari satelit yang mengorbit pada bumi. Jika modul mendapatkan sinyal dari satelit maka sistem dapat menentukan koordinat *Latitude* dan *Longitude*. Apabila tidak mendapatkan sinyal dari satelit, sistem akan mengulangi proses pencarian sinyal hingga mendapatkannya. Setelah modul GPS mendapatkan sinyal dari satelit data GPS (*Latitude*, *Longitude*) kemudian akan diproses untuk disimpan sebagai file rekaman dan pembuatan rute.



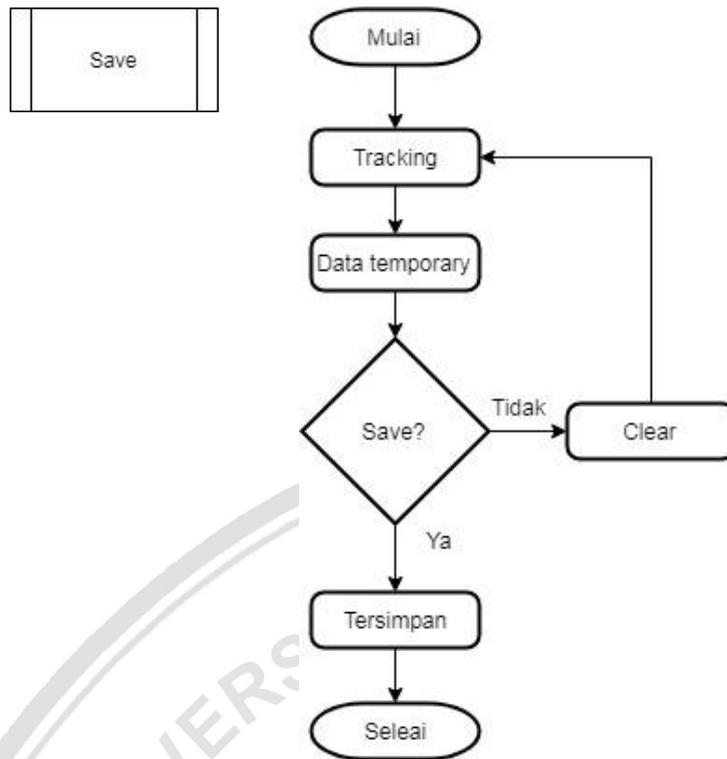
Gambar 5.9 Flowchart Sub-Program Tracking Data GPS

Pada sistem yang dirancang akan ditambah dengan algoritma kalman filter untuk mendapatkan nilai akurasi dari data yang diproses. Kalman filter akan mengestimasi nilai waktu awal (Prediksi) pembacaan data GPS untuk mendapatkan nilai waktu saat ini pembacaan sensor yang digunakan sebagai nilai koreksi sebagai hasil filter (*update*) untuk olah sistem. Data koordinat kemudian diPloting dalam grafik berupa garis hasil perjalanan yang dibedakan dengan warna biru untuk *Track* dan warna orange untuk garis *Trackback*.

d) Sub-program Save data

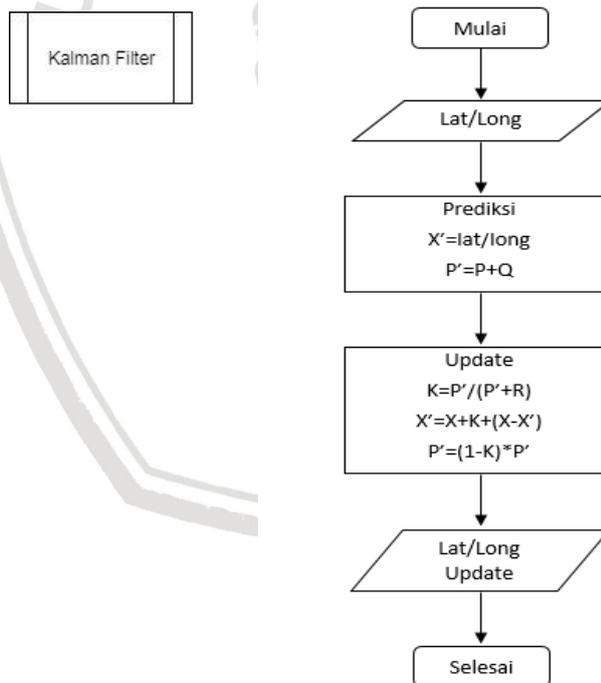
Pada perancangan *Save data*, sistem akan menyimpan data yang diperoleh pada proses *tracking* sesuai dengan keinginan pengguna. Data hasil *tracking* akan disimpan dalam sebuah file temporary. Data dapat disimpan ketika pengguna menekan tombol *Save* pada tampilan grap. Jika pengguna tidak menyimpan data maka, data akan dibersihkan pada saat pengguna kembali melakukan proses *tracking*, data lama yang tidak disimpan akan dihapus dan digantikan dengan data hasil *tracking* yang baru. Alur penyimpanan data seperti gambar berikut.





Gambar 5.10 Flowchart Sub-program Save Data

e) Sub-Program Kalman Filter



Gambar 5.11 Flowchart Sub-Program Kalman Filter

Pada perancangan sistem ini digunakan kalman filter sebagai penyaring data GPS. Dalam pengambilan data dari modul GPS yang digunakan, data yang diterima digunakan sebagai input pada proses pengolahan data, termasuk data yang

mengandung gangguan (*noise*). Gangguan tersebut terjadi akibat pengaruh perangkat elektronik lain di sekitar penggunaan sistem. Untuk meredam data gangguan, yang dijadikan input harus disaring terlebih dahulu dengan menggunakan kalman filter. Kalman filter akan mengestimasi keadaan suatu proses yang terjadi sebelum saat ini, dan yang akan datang. Proses penyaringan data kalman filter memiliki dua proses utama. Pertama kalman filter akan mengestimasi nilai koordinat sebelumnya (X) untuk mendapatkan nilai koordinat saat ini atau proses prediksi. Kemudian nilai koordinat sebelumnya (X') akan digunakan sebagai informasi untuk mendapatkan nilai saat ini atau proses *Update*. Kedua proses ini akan terus berjalan setiap sistem mendapatkan nilai input data koordinat dari GPS.

5.2 Implementasi

5.2.1 Implementasi Desain sistem GPS *Track Back*



Gambar 5.12 Implementasi Desain GPS *Track Back*

Pada implementasi perangkat keras secara keseluruhan yang telah dirancang, terdapat rangkaian komponen yang akan dipasang di dalam wadah yang sudah didesain untuk memperoleh bentuk secara tampilan dari alat yang dirancang. Hasil implementasi dari desain wadah yang dibuat adalah seperti gambar 5.9 berikut. Didalam wadah yang dibuat terdapat rangkaian keseluruhan komponen yang sudah digabungkan antar komponen yang meliputi sebagai berikut :

1. *Display LCD 800 x 480*
2. Modul *GPS U-blox Neo 6 m*
3. Kabel USB tipe B mini
4. Raspberry Pi 2B
5. Power Bank

a) Implementasi Display LCD 800 x 480

Pada implementasi Display LCD 800 x 480 pada sistem dijelaskan rangkaian perangkat keras untuk dapat berjalan sesuai dengan fungsi yang diharapkan sebagai berikut :



Gambar 5.13 Implementasi Rangkaian Display LCD 800 x 480

Rangkaian *display* LCD yang digunakan pada sistem menggunakan XPT2046 *Touch Controller 5inch HDMI Display* dengan resolusi 800 x 480 *Pixel* yang dapat diimplementasikan dengan *Raspberry Pi 2B*. LCD yang digunakan dibekali dengan 13 x 2 pin GPIO, port HDMI, port Mini USB, Backlight on/off, dan TFT *touch screen*. Untuk dapat bekerja pada *Raspberry Pi 2B*, LCD dihubungkan dengan *connector* HDMI, pin VCC, GND, MOSI, MISO, IRQ, CS, SCLK untuk mengaktifkan fungsi *touch screen*. sedangkan untuk menampilkan *display* menggunakan sambungan HDMI.

b) Implementasi Modul GPS Ublox Neo 6M

Pada implementasi perangkat keras digunakan module GPS *Neo 6M* sebagai komponen pengambil data *Longitude* dan *Latitude*. Module GPS *Ublox Neo 6M* dibekali dengan pin VCC, GND, RX dan TX. Module dapat bekerja pada tegangan 3,5 V pada *Raspberry Pi 2B*. Sebagai komunikasi untuk dapat mengirimkan data pin RX – TX pada modul disambungkan bersilang dengan GPIO RX – TX pada *Raspberry Pi 2B*. RX diartikan sebagai *Receiver* sedangkan TX diartikan sebagai *Transmitter*. Berikut adalah gambar implementasi perangkat keras modul *Ublox Neo 6M* :



Gambar 5.14 Implementasi Modul Ublox Neo 6M

Untuk dapat menjalankan modul GPS pada *Raspberry Pi 2B* dibutuhkan beberapa konfigurasi melalui *comand lineterminal* pada sistem operasi *Raspbian*

jessy. Langkah pertama adalah dengan menonaktifkan port *serial* pada *Raspberry Pi 2B*. kemudian masuk kedalam *cmdline* untuk mengubah baudrate dengan mengetikkan `sudo nano /boot/cmdline.txt` dan hapus `console=ttyAMA0,115200` kemudian `save`. Setelah konfigurasi dilakukan, instal `sudo apt-get install gpsd gpsd-clients`. Setelah instalasi selesai, kemudian yang awalnya baudrate 115200 di hapus, jalankan baudrate 9600 dengan cara mengetikkan `stty -F /dev/ttyAMA0 9600`, kemudian aktifkan `gpsd.sock` dengan `sudo gpsd /dev/ttyAMA0 -F /var/run/gpsd.sock` dan `gps` siap digunakan dengan perintah `cgps -s`.

5.2.2 Implementasi Perangkat Lunak

Pada Implementasi perangkat lunak dilakukan pembuatan program untuk melakukan pengambilan data kordinat (*Latitude, Longitude*), proses plot data GPS menjadi rekaman rute dengan pembahasan berdasarkan fungsi utama pada sistem yang dirancang. Dalam rancang bangun GPS track back pada rekaman rute pendakian ini yang menjadi pokok pembahasan adalah bagaimana sistem yang dirancang dapat melakukan proses perekaman rute berdasarkan data GPS. Pada implementasi proses perekaman yang menjadi bagian utama yang mempunyai kendali melakukan proses perekaman adalah fungsi *Track* dan *TrackBack* untuk menampilkan hasil perekaman rute pendakian.

a) Implementasi Library yang Digunakan

Pada sub bab implementasi sistem perekam rute pendakian, digunakan beberapa *library* dalam pembuatan program. *Library* yang digunakan antara lain *tkinter* yang digunakan untuk pembuatan *user interface* atau tampilan yang dapat menampilkan proses perekaman dengan menggunakan bahasa pemrograman *Python* pada *Raspberry Pi 2B*. kemudian digunakan *library matplotlib* yang digunakan untuk mengolah data GPS yang didapatkan untuk dirubah menjadi tampilan grafik koordinat dan.

Tabel 5.3 Library

No	Librarry
1	<code>import tkinter as tk</code>
2	<code>import matplotlib</code>

b) Implementasi Button Click Track

Pada bagian implementasi program *Track* ini dibuat proses fungsi *button click* dengan nama *Track* yang digunakan untuk melakukan perjalanan awal pada saat melakukan perekaman sebagai acuan rute kembali (*Back Track*). *Button click Track* dibuat untuk proses pengambilan data GPS berupa nilai koordinat *Latitude* dan *Longitude* dari sensor GPS yang digunakan untuk selanjutnya diplot menjadi garis kordinat perjalanan yang dapat dilihat dalam tampilan LCD *display*. Berikut kode program fungsi *Track* untuk pengambilan data GPS :



Tabel 5.4 Fungsi *Button Click Track*

NO	Kode Program Button Click Track
1
2	track_button = tk.Button(buttonframe, text="Track",
3	height=2, width=12, relief=tk.SOLID, bd=1,
4	command=self.track_clicked)
5	track_button.pack(side=tk.TOP, pady=4)
6	
7	def track_clicked(self):
8	global gps_stats, track_button
9	self.tr = threading.Thread(target=self.run_tracking)
10	self.tr.start()
12	track_button.configure(text="Stop", bg=conf.BLUE,
11	fg="white", command=self.stop_tracking)
12	gps_stats.config(text="Tracking Started",
13	fg=conf.BLUE)

Pada tabel diatas dibuat sebuah fungsi *button click Track* yang akan menjalankan proses pengambilan data GPS dan proses perekaman rute apabila fungsi *button click* diaktifkan dengan cara disentuh pada layar LCD. *Button click Track* akan mengarahkan kedalam fungsi *track_clicked*. Didalam fungsi ini terdapat beberapa konfigurasi yang akan merubah tampilan *button click Track* setelah ditekan, tampilan yang awalnya berupa *text="Track"* akan berubah menjadi *text="Stop"* dengan warna biru sebagai tanda fungsi *button click Track* yang dibuat berfungsi dan dapat digunakan.

c) Implementasi Proses *Run Tracking*

Pada bagian pengambilan data GPS dan perekaman dalam bentuk tampilan rute yang dilewati berdasarkan data koordinat GPS diperlukan sebuah fungsi yang dapat mengaktifkan sensor GPS untuk menangkap sinyal satelit untuk mendapatkan data GPS dan sekaligus melakukan perekaman pada sistem. Berikut tabel program fungsi *run_tracking* :

Tabel 5.5 Fungsi *Run Tracking*

NO	Kode Program Tracking
1	# -----
2	# -- Run tracking process --
3	# -----
4	def run_tracking(self):
5	global gps_stats, track_done
6	try:
7	self.running = True
8	self.gpsd = gps(mode=WATCH_ENABLE)
10	self.track_data = open(conf.TRACK_TMP_FILE, "w+")
11	while self.running:
12	self.gpsd.next()
13	latitude = self.gpsd.fix.latitude
14	longititude = self.gpsd.fix.longitude
15	if not math.isnan(latitude) and not
16	math.isnan(longititude) and latitude != 0.0 and
17	longititude != 0.0:



```

18         # Data format
19         data =
20         "{:.6f},{:.6f}".format(float(latitude), float(
21         longitude))
22         # Write data
23         self.track_data.write(str(data) + '\n')
24         self.track_data.flush()
25         os.fsync(self.track_data)
26         print(data)
27         gps_stats.config(text="Run Tracking",
28         fg=conf.BLUE)
29     else:
30         print "No Signal..."
31         gps_stats.config(text="No Signal",
32         fg=conf.RED)
33         time.sleep(2)
34
35     except (KeyboardInterrupt, SystemExit):
36         if self.running:
37             self.running = False
38             self.tr.join()
39             track_done = True
40         print "Tracking Stopped"

```

Dari program fungsi *run_tracking* ini dibuat alur proses perekaman mulai dari awal sensor GPS aktif. Fungsi *run_tracking* akan melakukan menjalankan tugasnya ketika *button click Track* aktif. Proses pertama yang dilakukan oleh fungsi *run_tracking* adalah mengambil data dari modul GPS dengan mengaktifkan *mode watch enable* dan kemudian membuka file baru untuk merekam data GPS yang didapat oleh modul GPS. Fungsi yang dibuat hanya mengambil data koordinat *Latitude* dan *Longitude* yang memiliki nilai selain 0, dengan format hingga 6 angka dibelakang koma {0.6F} format tersebut berdasarkan nilai DMS (*Degree Minute Secon*). Data koordinat yang didapat, selain disimpan dalam file temporary akan langsung diplot kedalam grafik dengan menggunakan fungsi *flush*. Dengan fungsi *flush* pengguna dapat melihat proses perekaman rute pada tampilan LCD *display* yang *diupdate* setiap 2 detik. Untuk mengetahui GPS mendapatkan sinyal dari satelit dan proses perekaman terus berjalan, dibuat sebuah pesan notifikasi "*run Tracking*" dengan warna biru yang muncul di bagian kanan bawah. Apabila GPS gagal mendapatkan data koordinat dari satelit untuk direkam maka, muncul pesan notifikasi "*No Signal*" dengan warna merah sebagai tanda gps sedang tidak mendapatkan data koordinat untuk direkam. Proses perekaman dan *Ploting* data koordinat GPS akan terus berulang hingga dihentikan dengan cara mengaktifkan *button click Stop*. Data koordinat GPS yang telah didapat dan ditulis dalam file *temporary* dapat disimpan secara permanen oleh pengguna diakhir proses perekaman.

d) Implementasi Button Click TrackBack

Pada sub bab implementasi program *TrackBack* dibuat *button click* dengan nama *TrackBack*. Setelah proses perekaman rute perjalanan dengan menggunakan fungsi *track* selesai, dapat dilakukan proses perekamna berikutnya dengan fungsi *button click TrackBack*. *Button click TrackBack* dibuat untuk proses

pengambilan data GPS berupa nilai koordinat *Latitude* dan *Longitude* dari sensor GPS yang kemudian data tersebut digunakan untuk proses perekaman perjalanan kembali (*Back Track*) yang selanjutnya diplot menjadi garis rute berdasarkan kordinat perjalanan yang dapat dilihat dalam tampilan LCD *display* dalam tampilan grafik. Berikut kode program fungsi *TrackBack* untuk pengambilan data GPS perjalanan kembali (*Back Track*):

Tabel 5.6 Fungsi *Button Click TrackBack*

NO	Kode Program Button Click Track Back
1	# TrackBack button
2	trackback_button = tk.Button(buttonframe,
3	text="TrackBack", height=2, width=12, relief=tk.SOLID,
4	bd=1, state="disabled", command=self.trackback_clicked)
5	trackback_button.pack(side=tk.TOP, pady=4)
6	
7	def trackback_clicked(self):
8	global gps_stats, trackback_button
9	self.tb =
10	threading.Thread(target=self.run_trackingback)
11	self.tb.start()
12	trackback_button.configure(text="Stop",
13	bg=conf.ORANGE, fg="white",
14	command=self.stop_trackingback)
15	gps_stats.config(text="TrackBack Started",
16	fg=conf.ORANGE)

Pada tabel diatas dibuat sebuah fungsi *button click TrackBack* yang akan menjalankan proses pengambilan data GPS dan proses perekaman rute kembali dengan cara menyentuh tombol *TrackBack* pada LCD *display* untuk mengaktifkan fungsi *button click TrackBack*. *Button click TrackBack* akan mengarahkan kefungsi *trackback_clicked*. Didalam fungsi ini terdapat beberapa konfigurasi yang akan merubah tampilan *button click TrackBack* yang mualanya berupa *text="TrackBack"* setelah ditekan atau diaktifkan akan berubah menjadi tampilan dengan format *text="Stop"* dengan warna oranye sebagai tanda fungsi *button click TrackBack* yang dibuat berfungsi dan dapat digunakan.

e) Implementasi fungsi Run Tracking Back

Pada bagian pengambilan data GPS dan perekaman dalam bentuk tampilan rute yang dilewati berdasarkan data koordinat GPS diperlukan sebuah fungsi yang dapat mengaktifkan sensor GPS untuk menangkap sinyal satelit untuk mendapatkan data GPS dan sekaligus melakukan perekaman pada sistem. Berikut tabel program fungsi *run_trackingback*:

Tabel 5.7 Fungsi *Run Track Back*

No	Kode Program Track Back
1	# -----
2	# -- Run trackingback process --
3	# -----
4	def run_trackingback(self):



```

5     global gps_stats, traceback_done
6     try:
7         self.running = True
8         print ""
9         print "TrackingBack Started"
10        print "-----"
11        self.traceback_data =
12        open(conf.TRACKBACK_TMP_FILE, "w+")
13        while self.running:
14            self.gpsd.next()
15            latitude = self.gpsd.fix.latitude
16            longitude = self.gpsd.fix.longitude
17            if not math.isnan(latitude) and not
18            math.isnan(longitude) and latitude != 0.0
19            and longitude != 0.0:
20                data =
21                "{:.6f},{:.6f}".format(float(latitude), f
22                loat(longitude))
23                self.traceback_data.write(str(data) +
24                '\n')
25                self.traceback_data.flush()
26                os.fsync(self.traceback_data)
27                print data
28                gps_stats.config(text="Run TrackBack",
29                fg=conf.ORANGE)
30            else:
31                print "No Signal..."
32                gps_stats.config(text="No Signal",
33                fg=conf.RED)
34                time.sleep(2)
35        except (KeyboardInterrupt, SystemExit):
36            if self.running:
37                self.running = False
38                self.tb.join()
39                traceback_done = True
40            print "Tracking Stopped"

```

Dari tabel fungsi *run_trackingback* ini dibuat proses yang memberikan perintah untuk melakukan pengambilan data GPS. Fungsi *run_trackingback* akan menjalankan tugasnya ketika *button click Track* diaktifkan dengan cara disentuh pada LCD *display* yang memiliki kemampuan *touch screen*. Proses pertama yang dilakukan oleh fungsi *run tracking back* adalah mengambil data dari modul GPS dengan mengaktifkan *mode= watch enable* dan kemudian membuka file *temporary* baru untuk menyimpan data koordinat yang diperoleh dari modul GPS dan nantinya akan disimpan sebagai file hasil perekaman data GPS. Fungsi yang dibuat hanya mengambil data koordinat *Latitude* dan *Longitude* yang memiliki nilai selain 0, dengan format hingga 6 angka dibelakang koma {0.6F} format tersebut berdasarkan nilai DMS (*Degree Minute Secon*). Data koordinat yang didapat selain disimpan dalam file *temporary* juga akan langsung diplot kedalam grafik dengan menggunakan fungsi *flush*. Dengan fungsi *flush* pengguna dapat melihat proses perekaman rute pada tampilan LCD *display* yang *diupdate* setiap 2 detik. Untuk mengetahui GPS mendapatkan sinyal dari satelit dan proses perekaman terus berjalan, dibuat sebuah pesan notifikasi "run Trackback" dengan

warna orange yang muncul dibagian kanan bawah. Apabila GPS gagal mendapatkan data koordinat dari satelit untuk direkam maka, muncul pesan notifikasi “No Signal” dengan warna merah sebagai tanda gps sedang tidak mendapatkan data koordinat untuk direkam. Proses perekaman dan *Ploting* data koordinat GPS akan terus berulang hingga dihentikan dengan cara mengaktifkan *button click Stop*. Data koordinat GPS yang telah didapat dan ditulis dalam file *temporary* dapat disimpan secara permanen oleh pengguna diakhir proses perekaman.

f) Implementasi Fungsi *Ploting* Data

Data GPS yang telah didapat melalui fungsi *track* dan *trackback* diatas kemudian akan diproses pada fungsi *plot* untuk dirubah menjadi grafik. Pada grafik yang terbaca pengguna dapat melihat rute perjalanan yang sedang direkam dalam bentuk rute hasil *Ploting* dari koordinat *Latitude* dan *Longitude*. Berikut tabel proses *Ploting* dari sistem :

Tabel 5.8 Fungsi *Ploting* Data

NO	Kode Program Plot data
1	def animate_a(i):
2	# Check if file exists, if not create files
3	if not os.path.exists(conf.TRACK_TMP_FILE) and not
4	os.path.exists(conf.TRACKBACK_TMP_FILE):
5	# Create track data files
6	trackData = open(conf.TRACK_TMP_FILE, "w")
7	trackData.write()
8	trackData.close()
9	# Create trackback data files
10	trackbackData = open(conf.TRACKBACK_TMP_FILE,
11	"w")
12	trackbackData.write()
13	trackbackData.close()
14	
15	trackData = open(conf.TRACK_TMP_FILE, "r").read()
16	trackbackData
17	=open(conf.TRACKBACK_TMP_FILE,"r").read()
18	trackList = trackData.split('\n')
19	trackbackList = trackbackData.split('\n')
20	tr_xList = []
21	tr_yList = []
22	for eachLine in trackList:
23	if len(eachLine) > 1:
24	lat, long = eachLine.split(',')
25	tr_xList.append(float(long))
26	tr_yList.append(float(lat))
27	tb_xList = []
28	tb_yList = []
29	for eachLine in trackbackList:
30	if len(eachLine) > 1:
31	lat, long = eachLine.split(',')
32	tb_xList.append(float(long))
33	tb_yList.append(float(lat))
34	plot_a.clear()
35	tracking_line, = plot_a.plot(tr_xList, tr_yList,
36	conf.BLUE, lw=2)



```
37   traceback_line, = plot_a.plot(tb_xList, tb_yList,
36   conf.ORANGE, lw=2)
```

Data GPS yang diperoleh dari hasil *tracking* dan *trackback* selanjutnya diproses untuk ditampilkan pada LCD *display*. Proses diawali dengan sistem akan memastikan file *temporary Track* dan *TrackBack* dalam keadaan tidak menerima input data GPS. Jika dalam file sudah dipastikan kosong maka, file *temporry Track* dan *trackback* akan menerima data GPS untuk ditulis. File yang pertama kali terisi adalah file *Track* yang digunakan sebagai acuan rute awal pendakian sebelum melakukan perjalanan kembali (*Back Track*). Sistem akan membaca data GPS dalam file *temporarry* secara berurutan dimulai dari file *temporarry track* kemudian *trackback*. Data koordinat GPS *latitude* dan *longitude* yang berada didalam file kemudian dibaca oleh sistem untuk dirubah menjadi grafik yang menggambarkan rute perjalanan yang sedang direkam secara berurutan sesuai masukan data yang diterima dari modul GPS. Garis rute yang dibuat dari hasil perjalanan dalam grafik dibedakan menjadi dua garis warna yaitu garis rute untuk hasil *Track* dibuat dengan warna biru sedangkan untuk *TrackBack* dibuat dengan warna orenge. Tujuan dari perbedaan warna agar menjadi pembeda antara grafik rute hasil *Track* dan *TrackBack*.

g) Implementasi Kalman Filter

Pada rancang bangun GPS *Back Track* pada rekaman rute pendakian ini digunakan algoritma kalman filter sebagai algoritma filter data untuk meminimalisir nilai gangguan pada data GPS yang digunakan sebagai data masukan proses perekaman. Kalman filer memiliki dua proses untuk dapat memprediksi nilai dari sensor GPS. Proses prediksi merupakan langkah pertama untuk mendapatkan nilai prediksi kesalahan sensor yang akan menjadi umpan balik dalam proses *measurement update*. Hasil dari implementasi kalman filter dijadikan sebagai perbandingan hasil rekaman yang dilakukan tanpa menggunakan filter data. Implementasi kalman filter pada sistem dijabarkan pada tabel 5.9 berikut :

Tabel 5.9 Implementasi Kalman Filter

NO	Kode Program Kalman Filter
1	# inisialisasi kalman filter latitude
2	Qlat = 0 #inisialisasi Q latitude
3	Rlat = 1 #inisialisasi R latitude
4	xhatlat = 0 #inisialisasi xhat latitude
5	Plat = 0.00001 #inisialisasi P latitude
6	
7	# inisialisasi kalman filter longitude
8	Qlong = 0 #inisialisasi Q longitude
9	Rlong = 1 #inisialisasi R longitude
10	xhatlong = 0 #inisialisasi xhat longitude
11	Plong = 0.00001 #inisialisasi P longitude
12	
13	while self.running:
14	# gpsd func
15	self.gpsd.next()
16	# Get fix latitude and longitude



```

17 latitude = self.gpsd.fix.latitude
18 longitude = self.gpsd.fix.longitude
19 # Kalman filtering for latitude
20 xhatlatbef = xhatlat # X(k-1) latitude
21 Platbef = Plat + Q #P(k-1)+Q latitude
22
23 Klat = Platbef / (Platbef + Rlat) #K latitude
24 xhatlat = xhatlatbef + Klat * (latitude-
25 xhatlatbef) #xhat latitude update
26 Plat = (1-Klat)*Platbef #P latitude baru update
27 # Kalman filtering for longitude
28 xhatlongbef = xhatlong # X(k-1) longitude
29 Plongbef = Plong + Q #P(k-1)+Q longitude
30
31 Klong = Plongbef / (Plongbef + Rlong) #K longitude
32 xhatlong = xhatlongbef + Klong * (longitude-
33 xhatlongbef) #xhat longitude update
34 Plong = (1-Klong)*Plongbef #P longitude baru
35 update

```

Dalam penelitian ini digunakan Kalman filter yang berfungsi sebagai estimator rata – rata error data GPS dengan membandingkan hasil pengukuran data waktu sebelumnya dan data pengukuran yang sedang berlangsung. Kalman filter menggunakan konsep dengan memprediksi (*Time Update*) untuk mendapatkan nilai estimasi dari waktu sebelumnya untuk nilai yang akan datang (*Measurement Update*) kemudian nilai yang sudah didapatkan akan menjadi *correct* dan diperbarui untuk digunakan kembali sebagai umpan balik nilai estimasi untuk *time update* dengan harapan akan mendapat nilai yang lebih akurat. Dalam proses prediksi data (*Time Update*) digunakan persamaan sebagai berikut.

$$\hat{X} = \hat{X}(k - 1)$$

Dan

$$P_{bef} = P + Q$$

Keterangan :

\hat{X} = nilai awal data GPS (*latitude, Longitude*)

P_{bef} = *covariance* sebelum diupdate

Q = estimasi error

Persamaan diatas digunakan untuk menentukan nilai X dan P yang akan digunakan sebagai umpan balik dalam proses selanjutnya (*measurement update*). Dalam proses perhitungan diatas nilai sensor (\hat{X}) akan dikalikan dengan hasil *index matrix* (k) dikurangi 1. Serupa dengan \hat{X} , untuk mendapatkan nilai P juga akan diproses sama dengan penentuan nilai \hat{X} . Kemudian setelah prediksi nilai awal \hat{X} dan P didapat, nilai tersebut akan digunakan sebagai umpan balik untuk mendapatkan nilai akhir hasil filter dengan melalui proses *measurement Update* dengan menggunakan persamaan sebagai berikut.



$$K = \frac{P_{bef}}{(P_{bef} + R)}$$

$$X = \hat{X} + K + (lat\ or\ long - \hat{X}_{bef})$$

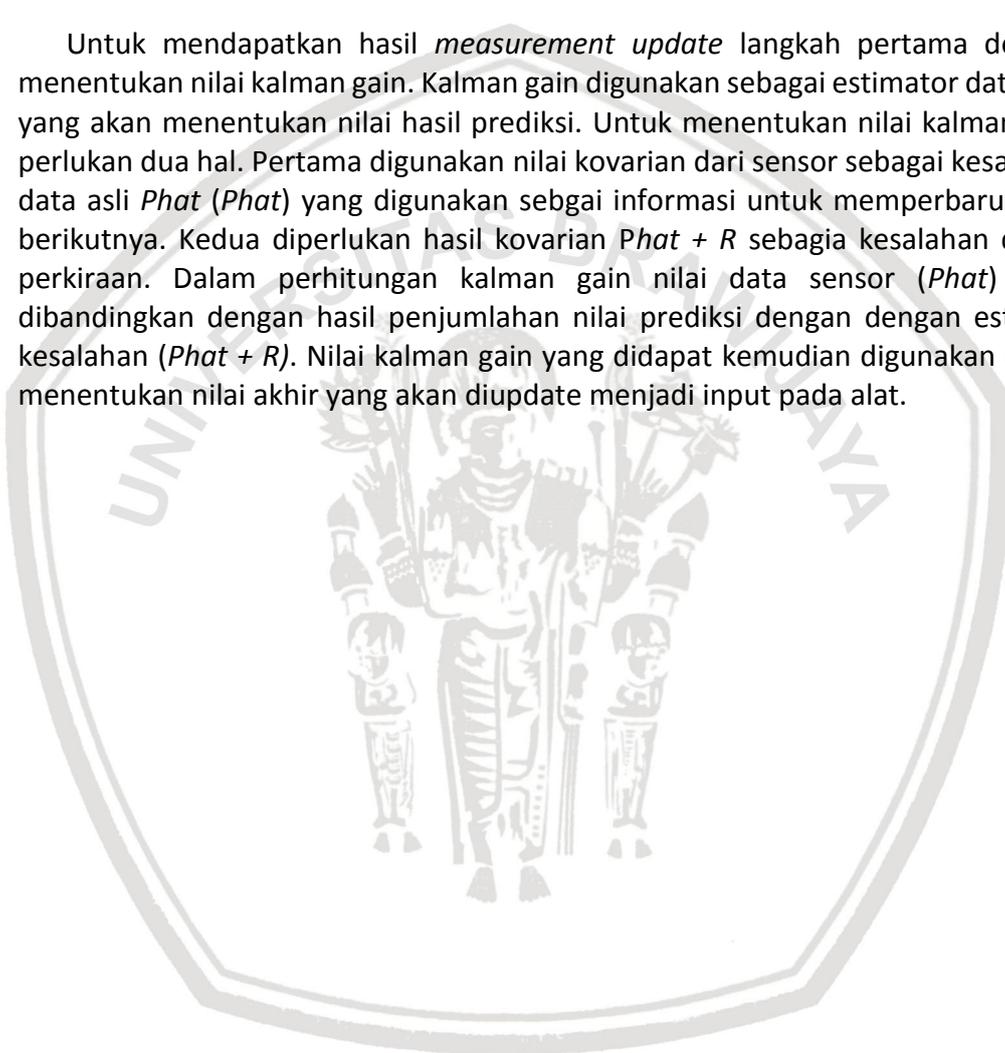
$$P = (1 - K_{lat\ or\ long}) \times P_{bef}$$

Keterangan :

K = Kalman Gain

R = perkiraan *error*

Untuk mendapatkan hasil *measurement update* langkah pertama dengan menentukan nilai kalman gain. Kalman gain digunakan sebagai estimator data GPS yang akan menentukan nilai hasil prediksi. Untuk menentukan nilai kalman gain perlukan dua hal. Pertama digunakan nilai kovarian dari sensor sebagai kesalahan data asli *Phat* (*Phat*) yang digunakan sebagai informasi untuk memperbarui data berikutnya. Kedua diperlukan hasil kovarian *Phat + R* sebagai kesalahan dalam perkiraan. Dalam perhitungan kalman gain nilai data sensor (*Phat*) akan dibandingkan dengan hasil penjumlahan nilai prediksi dengan dengan estimasi kesalahan (*Phat + R*). Nilai kalman gain yang didapat kemudian digunakan untuk menentukan nilai akhir yang akan diupdate menjadi input pada alat.



BAB 6

PENGUJIAN DAN ANALISIS

Pengujian alat *GPS Back Track* ini dilakukan dalam beberapa tahap diantaranya adalah pengujian fungsional dari alat yang sudah dirancang sebelumnya dan pengujian performa dari sistem yang telah dirancang untuk mengetahui kemampuan sistem saat digunakan.

6.1 Pengujian Nilai Akurasi Modul GPS Terhadap Nilai *Google Maps*

Dalam rancang bangun *GPS back track* pada rekaman rute, digunakan modul *GPS Ublox Neo 6M* sebagai pengolah data GPS. Pengujian akurasi modul GPS dalam akuisis data koordinat dilakukan dengan menggunakan *Raspberry Pi 2B* sebagai pemroses data GPS dengan melakukan percobaan tracking data GPS pada lokasi yang berbeda. Modul GPS dan *Google Maps* akan mengakuisis data dan menampilkan hasilnya pada LCD. Nilai Modul GPS dan *Google Maps* akan dibandingkan sehingga diperoleh selisih hasil dari nilai Modul GPS dan nilai *Google Maps* pada android.

6.1.1 Tujuan Pengujian

Tujuan pengujian dengan cara melakukan percobaan tracking data koordinat dilakukan guna mengetahui nilai selisih koordinat data GPS dari modul GPS yang dibandingkan dengan nilai koordinat yang terbaca pada *Google Maps* android sehingga dapat diketahui presentase *error* dari masing masing perangkat yang digunakan pada pengujian.

6.1.2 Prosedur Pengujian

Berikut merupakan prosedur pengujian dengan melakukan percobaan nilai koordinat pada modul GPS terhadap nilai koordinat *Google Maps* android.

- a. Menghubungkan rangkaian modul GPS, *Raspberry Pi 2B* dan LCD pada sumber daya.
- b. Membuka *Google Maps* pada android dan memastikan mode GPS aktif.
- c. Jalankan modul GPS melalui terminal pada sistem operasi *Raspberry*.
- d. Tunggu hingga modul GPS mendapat sinyal satelit dan dapat menampilkan nilai koordinat.
- e. Mendokumentasikan nilai yang ditampilkan oleh modul GPS.
- f. Memasukkan nilai koordinat yang terbaca oleh modul GPS pada pencarian di *Google Maps*.
- g. Mendokumentasikan dan mencatat hasil pencarian lokasi pada *Google Maps*.

- h. Merubah nilai koordinat yang terbaca oleh Google Maps dari *Degree Minute Secon (DMS)* menjadi desimal dengan rumus persamaan

$$Latitude/Longitude = Degree + \left(\frac{Minute}{60}\right) + \left(\frac{second}{3600}\right)$$

Berikut merupakan persamaan yang digunakan sebagai penghitung presentase selisih *error* dari setiap nilai koordinat data GPS.

$$Presentase\ error = \frac{Nilai\ modul\ GPS - Nilai\ Google\ maps}{Nilai\ Google\ maps} \times 100$$

6.1.3 Hasil dan Analisis Akurasi Nilai *Latitude*

Berdasarkan langkah-langkah yang telah dilakukan dalam percobaan tracking data koordinat untuk mengetahui tingkat akurasi GPS didapatkan hasil sebagai berikut.

Tabel 6.1 Percobaan Modul GPS Dengan *Google Maps (Latitude)*

No	Modul GPS	Google maps	Presentase error
1	7.947955	7.947944	0.0001384%
2	7.948091	7.948083	0.0001006%
3	7.946234	7.946222	0.0001510%
4	7.945407	7.945416	0.0001132%
Rata – rata error			0.0001258%

Dengan menggunakan persamaan presentase *error* didapatkan hasil nilai *error* dari setiap nilai koordinat *Latitude* GPS yang didapatkan. Hasil perhitungan presentase *error* dari data koordinat *Latitude* dengan melakukan 4 kali percobaan di lokasi yang berbeda, seperti yang disajikan pada tabel 6.1.

$$Presentase\ error = \frac{Nilai\ modul\ GPS - Nilai\ Google\ maps}{Nilai\ Google\ maps} \times 100$$

$$Rata - rata\ error = \frac{0.0005033}{4} \times 100$$

$$Akurasi\ GPS = 100 - 0.0001258\%$$

$$Akurasi\ GPS = 99.999874\%$$

Presentase *error* diperoleh melalui pembacaan nilai koordinat modul GPS yang dibandingkan dengan nilai koordinat yang terbaca oleh *Google Maps* pada perangkat android. Untuk mendapatkan presentase *error* dari data koordinat dilakukan dengan cara melakukan oprasi pengurangan data koordinat *Latitude*



Modul GPS terhadap data koordinat *Latitude* pada Google Maps perangkat android. kemudian hasil pengurangan dibagi dengan nilai koordinat *Latitude* dari Google Maps perangkat android dan dikalikan dengan nilai 100. Berdasarkan nilai rata – rata koordinat *Latitude* yang didapat dari modul GPS dan Google maps didapatkan nilai rata – rata presentasi *error* sebesar 0.0001258% dan nilai akurasi GPS 99.999874%.

6.1.4 Hasil dan Analisis Akurasi Nilai *Longitude*

Tabel 6.2 Percobaan Modul GPS Dengan Google Maps (*Longitude*)

No	Modul GPS	Google maps	Presentase error
1	112.603676	112.603666	0.0000000888 %
2	112.604759	112.604769	0.0000000888%
3	112.611050	112.611055	0.0000000444%
4	112.618608	112.601944	0.000147%
Rata – rata error			0.000037%

Dengan menggunakan persamaan presentase *error* didapatkan hasil nilai *error* dari setiap nilai koordinat *Longitude* GPS yang didapatkan. Hasil perhitungan presentase *error* dari data koordinat *Longitude* dengan melakukan 4 kali percobaan pada tempat yang berbeda, seperti yang disajikan pada tabel 6.2.

$$\text{Presentase error} = \frac{\text{Nilai modul GPS} - \text{Nilai Google maps}}{\text{Nilai Google maps}} \times 100$$

$$\text{Rata – rata error} = \frac{0.000148}{4}$$

$$\text{Akurasi GPS} = 100 - 0.000037\%$$

$$\text{Akurasi GPS} = 99.99996\%$$

Presentase *error* diperoleh melalui pembacaan nilai koordinat modul GPS yang dibandingkan dengan nilai koordinat yang terbaca oleh Google Maps pada perangkat android. Untuk mendapatkan presentase *error* dari data koordinat dilakukan dengan cara melakukan oprasi pengurangan data koordinat *Longitude* Modul GPS terhadap data koordinat *Longitude* pada Google Maps perangkat android, kemudian hasil pengurangan dibagi dengan nilai koordinat *Longitude* dari Google Maps perangkat android dan dikalikan dengan nilai 100. Berdasarkan nilai rata – rata koordinat *Longitude* yang didapat dari modul GPS dan Google maps didapatkan nilai rata – rata presentasi *error* sebesar 0.000037% dan akurasi GPS 99.99996%.



6.1.5 Hasil dan Analisis Modul GPS Terhadap Waktu akuisisi Data

Tabel 6.3 Percobaan Modul GPS Terhadap Waktu Akuisisi Data

No	Lokasi Akuisisi	Koordinat	Waktu (s)
1	Jl. Joyo Suko	7.947955, 112.603676	100
2	Jl. Raya Candi	7.948091, 112.604759	64
3	Jl. Kerto Raharjo	7.946224, 112.611050	38
4	Jl. Soekarno-Hatta	7.945407, 112.618608	714
Rata – rata waktu akuisisi data			229

Dari hasil pengujian modul GPS Ublox Neo 6 M yang digunakan pada sistem dengan melakukan percobaan perhitungan lama waktu GPS memperoleh sinyal didapatkan hasil, lama waktu yang dibutuhkan modul GPS untuk mendapatkan sinyal satelit dalam melacak koordinat lokasi yang dilakukan di 4 tempat yang berbeda seperti tabel 6.3. untuk mendapatkan presentasi rata – rata waktu pengambilan data digunakan persamaan sebagai berikut.

$$\text{Rata – rata waktu} = \frac{\text{Jumlah seluruh waktu}}{\text{banyak data}}$$

$$\text{Rata – rata} = \frac{916}{4}$$

$$\text{Rata – rata} = 229$$

Rata – rata waktu pengambilan data GPS diperoleh dari pertamakali modul GPS diaktifkan hingga mendaptkan sinyal dan menerima nilai koordinat dari satelit. Untuk menghitung rata – rata waktu yang diperlukan modul GPS untuk menerima data dengan cara menjumlahkan seluruh waktu yang didapat dari modul GPS kemudian membaginya dengan banyak data sesuai jumlah lokasi dilakukannya pengujian. Jumlah waktu yang diperlukan modul GPS menerima sinyal satelit dari empat kali pengujian di lokasi yang berbeda selama 916 detik (s). setelah mendapatkan jumlah keseluruhan waktu, maka didapatkan hasil rata – rata waktu yang dibutuhkan modul GPS untuk menampilkan data adalah selama 229 detik (s). Lama waktu proses pengambilan data pada modul GPS disebabkan oleh pengaruh lingkungan disekitar lokasi percobaan yang terdapat banyak bangunan tinggi atau bangunan yang terbuat dari beton. Dari hasil analisis dapat disimpulkan modul GPS yang akan bekerja maksimal ketika berada diruanagn terbuka tanpa memiliki banyak penghalang seperti bangunan permanen atau bangunan yang memiliki lebih dari 1 lantai.

6.2 Pengujian dan Analisis Terhadap Proses Perekaman Rute

Dalam perancangan GPS *back track* pada rekama rute pendakian dibuat menu *Track*, *TrackBack*, dan *Save* yang ditampilkan pada halaman *Graph*, dalam bentuk fungsi *button click*. Fungsi dari menu tersebut digunakan untuk menjalankan

proses perekaman rute, seperti pengambilan data GPS dan melakukan proses penyimpanan data GPS sebagai hasil rekaman rute.

6.2.1 Tujuan pengujian

Pengujian terhadap proses perekaman rute dilakukan dengan tujuan dapat mengetahui menu *Track*, *TrackBack*, dan *Save* dapat melakukan proses pengolahan data GPS menjadi garis rute yang dilalui dengan menampilkannya pada layar LCD dan dapat menyimpan file hasil perekaman.

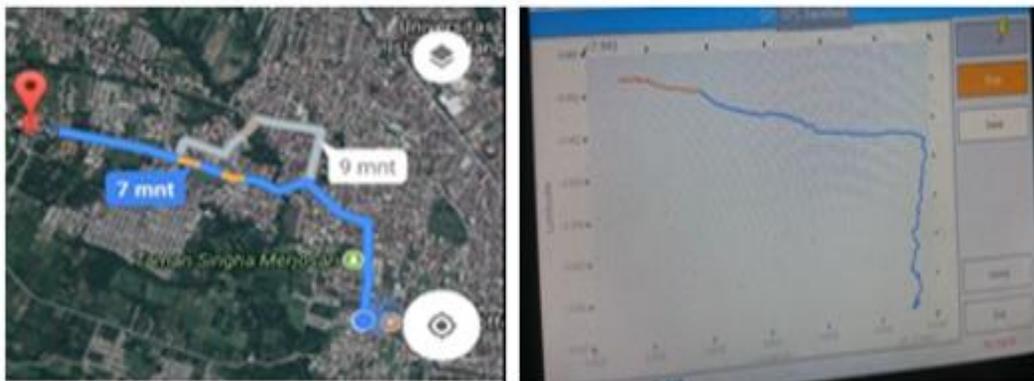
6.2.2 Prosedur pengujian

Berikut ini merupakan langkah – langkah yang dilakukan dalam proses perekaman.

1. Menyalakan alat yang telah dibuat
2. Menjalankan program tampilan yang telah dibuat.
3. Membuka *Google Maps* untuk menentukan rute tujuan sebagai acuan rute yang akan dilewati.
4. Masuk dalam tampilan *Graph* dengan klik menu *New* pada tampilan *Home*.
5. Klik menu *Track* untuk melakukan perekaman rute awal.
6. Klik menu *TrackBack* untuk melakukan perekaman rute kembali ke posisi awal berdasarkan rute *track*.
7. Mendokumentasikan hasil perekaman.
8. Klik menu *Save* untuk menyimpan hasil perekaman.

6.2.3 Hasil dan Analisa Proses Perekaman Rute

Hasil pengujian pada proses perekaman rute seperti Gambar 6.1.



Google Maps

Hasil Rekaman

Gambar 6.1 Hasil Proses Perekaman

Dalam pengujian proses perekaman dilakukan penentuan rute tujuan yang akan dilalui dengan menggunakan *Google Maps*. Rute perjalanan yang muncul pada *Google Maps* dijadikan acuan rute yang akan direkam yang dapat diketahui

hasil rekaman akan memiliki tampilan yang kurang lebih sama dengan rute yang ditampilkan oleh *google maps*. Dari hasil pengujian proses perekaman rute perjalanan, menu *Track* dapat menjalankan fungsi pengambilan dan mengolah data GPS dalam bentuk garis rute perjalanan yang ditandai dengan garis warna biru. Kemudian menu *TrackBack* dapat mengolah data GPS dalam bentuk tampilan garis rute berwarna orange. Berdasarkan pengujian peroses perekaman diperoleh hasil, fungsi *Track* dan *Trackback* yang dirancang pada rancang bangun GPS *back track* dapat berjalan sesuai dengan fungsi masing – masing dan mampu merekam rute perjalanan yang menyerupai rute pada *google maps*.

6.3 Pengujian dan Analisis Kinerja Fungsi Open File

Dalam rancang bangun GPS *back track* pada rekaman pendakian dibuat sebuah fungsi *button click Open*. *Button click Open* digunakan sebagai fungsi yang mengarahkan pengguna ke direktori penyimpanan file hasil rekaman untuk membuka file rekaman yang telah tersimpan sebelumnya.

6.3.1 Tujuan Pengujian

Pengujian dilakukan dengan tujuan dapat mengetahui fungsi *button click Open* yang dibuat dapat membuka file yang telah disimpan. Sehingga dapat melihat hasil rekaman rute yang sudah disimpan sebelumnya.

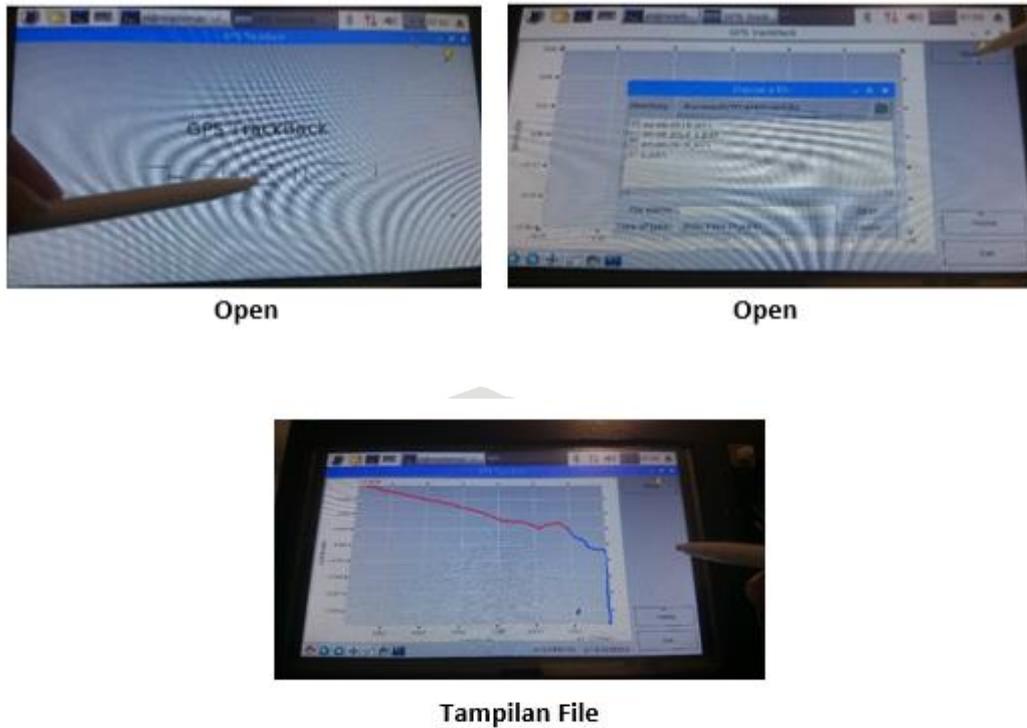
6.3.2 Prosedur Pengujian

Berikut ini merupakan langkah – langkah dalam pengujian fungsi program tampilan

1. Mengaktifkan rangkaian alat yang telah dibuat.
2. Jalankan program yang telah dibuat.
3. Klik menu *Open* di dalam *Home*.
4. Klik menu *Open* di dalam halaman *Graph* untuk memilih file.
5. Pilih file rekaman yang tersimpan yang ingin dibuka.
6. Klik open.
7. Mendokumentasikan tampilan.

6.3.3 Hasil dan Analisis Proses Open File Rekaman

Hasil pengujian dapat dilihat pada Gambar 6. 2



Gambar 6.2 Hasil Pengujian Fungsi *Open File*

Dengan dilakukannya pengujian terhadap fungsi *Open*, perancangan yang diimplementasikan pada sistem rekaman dapat bekerja sesuai dengan fungsinya. *Button click Open* dapat membuka file hasil perekaman yang tersimpan pada sistem dengan menampilkan *box dialog* direktori lokasi file terimpan. Dari hasil pengujian dapat dianalisis bahwa data yang diolah oleh sistem dapat disimpan dan dibuka kembali dengan menggunakan fungsi *button click Open*. Sehingga hasil rekaman yang telah dilakukan sebelumnya dapat dilihat kembali pada tampilan LCD.

6.4 Pengujian dan Analisis Nilai *tracking* dengan Kalman Filter

Untuk mengetahui Kalman filter yang dirancang pada sistem dapat bekerja menyaring data koordinat, maka dilakukan pengujian performa Kalman filter pada sistem dengan cara mengambil beberapa data sampel dari data koordinat yang didapat dari proses perekaman rute. Sehingga diketahui hasil selisih perbandingan koordinat *tracking* yang tidak menggunakan Kalman filter dan menggunakan Kalman filter.

6.4.1 Tujuan Pengujian

Pengujian dilakukan dengan tujuan Kalman filter yang digunakan pada sistem dapat bekerja dan mendapatkan hasil selisih nilai koordinat yang dilakukan pada posisi yang sama.

6.4.2 Prosedur Pengujian

Berikut ini merupakan prosedur yang dilakukan dalam proses pengujian :

1. Jalankan program untuk melakukan *tracking*.
2. lakukan *tracking singkat* untuk mendapatkan nilai koordinat.
3. Mengambil beberapa data sampel dari data koordinat yang diperoleh dalam perekaman.
4. Simpan hasil dan analisis hasil *tracking* yang diperoleh.

6.4.3 Hasil dan Analisis Selisih Nilai Koordianat

Berikut tabel hasil *tracking* yang diperoleh

Tabel 6.4 Selisih Koordinat Hasil Tracking

No	Tanpa Kalman	Kalman Filter	Selisih Latitude	Selisih Longitude
1	-7.947976, 112.603509	-7.947966, 112.603518	0.000010	0.000010
2	-7.947978, 112.603507	-7.947969, 112.603517	0.000011	0.000010
3	-7.947980, 112.603504	-7.947971, 112.603514	0.000011	0.000010
4	-7.947984, 112.603512	-7.947974, 112.603522	0.000010	0.000010
5	-7.947988, 112.603517	-7.947978, 112.603527	0.000010	0.000010
6	-7.947987, 112.603515	-7.947978, 112.603525	0.000011	0.000010
7	-7.947984, 112.603543	-7.947974, 112.603553	0.000010	0.000010
8	-7.947992, 112.603530	-7.947982, 112.603540	0.000010	0.000010
9	-7.947991, 112.603533	-7.947981, 112.603542	0.000010	0.000011
10	-7.947987, 112.603538	-7.947978, 112.603548	0.000011	0.000010
Rata – rata selisih hasil tracking			0.000010	0.000010

Dari pengujian yang dilakukan untuk mengetahui Kalman filter dapat melakukan proses filter data dan mengetahui selisih nilai koordinat hasil *tracking* diperoleh hasil Kalman filter yang digunakan dapat melakukan proses filtering data koordinat yang diterima dari modul sensor yang digunakan. Kemudian dari hasil *tracking* koordinat tanda menggunakan Kalman filter dan menggunakan kalman filter diperoleh rata-rata selisih koordinat latitude sebesar 0.000010 dan koordinat longitude sebesar 0.000010.

6.5 Pengujian dan Analisis Tracking Data

6.6 Kalman Filter

Kalman filter pada rancang bangun GPS *back track* pada rekaman rute pendakian digunakan sebagai pembanding. Hasil rekaman data GPS yang dibuat tanpa filter data akan dibandingkan dengan data hasil rekaman yang menggunakan kalman filter sebagai filter data GPS. Dari proses filter yang dilakukan diharap memperoleh hasil *Tracking* GPS dengan hasil yang lebih akurat.

Sehingga hasil yang diperoleh kalman filter dapat menjadi pembandingan hasil rekaman tanpa filter data.

6.6.1 Tujuan Pengujian

Pengujian dilakukan dengan tujuan mendapatkan hasil perbandingan antara data GPS murni tanpa filter dengan data GPS yang diolah menggunakan kalman filter.

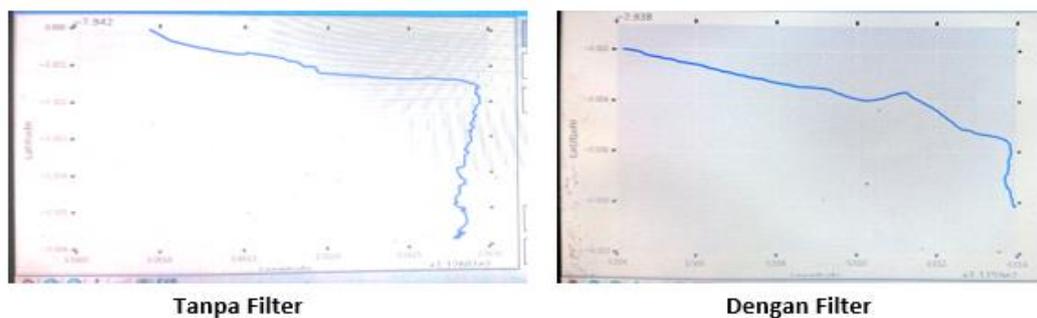
6.6.2 Prosedur Pengujian

Berikut ini merupakan langkah – langkah dalam pengujian filtering data sebagai pembandingan :

1. Menjalankan program yang telah ditambahkan fungsi kalman filter .
2. Tekan *button click track* dan lakukan *tracking data* dengan rute yang suda disiapkan.
3. Dokumentasikan hasil perekaman dengan menggunakan kalman filter.
4. Tekan *button click trackback* untuk melakukan perekaman perjalanan kembali.
5. Mendokumentasikan hasil perekaman.
6. Menganalisis hasil rekaman tanpa filtering data dengan hasil perekaman yang menggunakan kalman filter

6.6.3 Hasil dan Analisa *Tracking Data* Menggunakan Kalman Filter

Hasil pengujian dapat dilihat pada gambar 6. 3



Gambar 6.3 Hasil Tracking Data GPS

Dari pengujian *tracking data* GPS dengan menggunakan kalman filter diperoleh hasil, garis rute yang terbentuk terlihat lebih halus dan rapi dibandingkan dengan garis rute yang diambil tanpa melalui proses filterisasi menggunakan kalman filter. Rute yang terbentuk dari data GPS tanpa melalui filterisasi terlihat lebih berantakan dan terdapat gangguan yang sangat terlihat. Perbedaan tersebut disebabkan karena data yang diterima sistem sebagai masukan tanpa filter merupakan data murni yang masih bercampur dengan gangguan (*nois*). Berbeda dengan data yang diproses melalui filter terlebih dahulu. Data GPS masih mengandung gangguan akan difilter sehingga mendapat hasil yang mendekati akurat.

6.7 Pengujian dan Analisis perekaman dengan acuan Google maps

Untuk mengetahui jika alat yang dirancang dapat bekerja sesuai dengan fungsi yang diinginkan maka, dilakukan pengujian fungsi pokok dari alat GPS perekam rute yang telah dibuat. Pengujian ini dilakukan dua kali perjalanan pulang pergi (*Track* dan *Back Track*) dengan cara menentukan rute perjalanan yang akan ditempuh dengan menggunakan aplikasi *Google Maps* pada perangkat android. Rute yang tampil pada *maps* menjadi acuan rute yang akan dibuat dan direkam oleh alat dan diketahui hasil rekaman apakah sudah menyerupai rute yang telah ditentukan sebelumnya pada *maps*.

6.7.1 Tujuan Pengujian

Pengujian dilakukan dengan tujuan alat yang dibuat dapat menjalankan fungsi utama perekaman jalur berdasarkan nilai koordinat GPS dan merubah nilai koordinat menjadi grafik rute perjalanan.

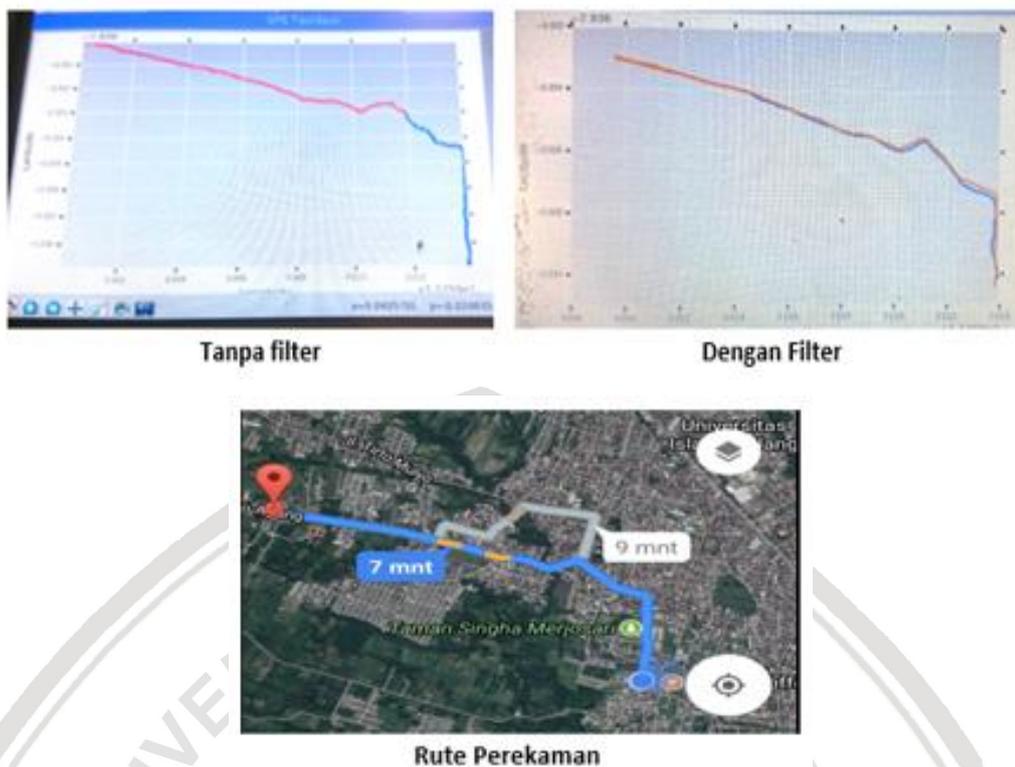
6.7.2 Prosedur Pengujian

Berikut ini merupakan langkah – langkah dalam pengujian yang dilakukan.

1. Mengaktifkan rangkaian alat yang telah dibuat.
2. Jalankan program yang telah dibuat.
3. Klik menu *New*.
4. Lakukan perjalanan ke titik tujuan dengan klik *Track*.
5. Lakukan perjalanan ke titik awal dengan klik *Trackback*.
6. Klik save untuk menyimpan hasil rekaman kemudian *Exit*.
7. Jalankan program yang menggunakan kalman filter untuk filter data.
8. Klik menu *New*.
9. Lakukan perjalanan ke titik tujuan dengan klik *Track*.
10. Lakukan perjalanan ke titik awal dengan klik *Trackback*.
11. Simpan hasil rekaman dengan klik *Save* kemudian *Exit*.
12. Melakukan analisis hasil rekaman dari dua program yang berbeda.

6.7.3 Hasil dan Analisis Proses Open File Rekaman

Hasil pengujian dapat dilihat pada gambar 6. 4



Gambar 6.4 Hasil Pengujian *Track* dan *Back Track* Dengan Acuan *Google Maps*.

Berdasarkan pengujian yang telah dilakukan diperoleh hasil, fungsi *track* dan *track back* yang dibuat untuk melakukan perekaman data koordianat GPS dapat melakukan perekaman dan menampilkan hasil perekaman pada LCD *display*. Hasil rekaman menggunakan program tanpa penambahan filter data terlihat garis rute yang terekam memperlihatkan garis rute yang lebih kasar. Hal tersebut dikarenakan data gangguan (*nois*) yang diterima oleh modul GPS sebagai masukan ikut tersimpan dalam file rekaman. Sehingga, garis rute yang terbentuk memiliki loncatan – loncatan kecil yang menyebabkan tampilan garis terlihat kasar. Berbeda dengan hasil perekaman yang dilakukan menggunakan filter, garis hasil rekaman terlihat lebih halus. Hal tersebut dikarenakan dalam proses perekaman data koordinat dari modul GPS yang masih memiliki gangguan, disaring terlebih dahulu sebelum disimpan sebagai hasil rekaman. Dalam proses penyaringan data digunakan kalman filter sebagai penyaring data. Kalman filter menggunakan perkiraan gangguan (*error*) sebagai umpan balik untuk memprediksi hasil perhitungan yang mendekati lebih akurat. Dari hasil pengujian diatas dapat disimpulkan penggunaan filter data dalam perekaman diperoleh hasil perekaman rute yang lebih menyerupai rute acuan pada *google maps* yang digunakan.

BAB 7 KESIMPULAN DAN SARAN

7.1 Kesimpulan

Dari tahapan yang telah dilakukan secara keseluruhan didapatkan hasil pengujian dan hasil analisis dapat disimpulkan sebagai berikut :

1. Rancang bangun *GPS Back Track* menggunakan Raspberry Pi 2B sebagai mikrokontroler, *display LCD 800 x 480*, modul GPS Ublox Neo 6 M. Modul GPS *Ublox Neo 6 M* berfungsi mengolah data GPS sehingga didapatkan nilai koordinat yang digunakan sebagai penentu posisi pengguna, dan digunakan untuk memantau pergerakan pengguna. *Display LCD 800 x 480* yang digunakan dapat menampilkan proses perekaman rute yang dilakukan. Raspberry Pi dapat mengolah data GPS tanpa menggunakan kalman filter dan dapat mengolah data dengan menggunakan kalman filter.
2. *GPS Back Track* dapat merekam rute perjalanan pengguna dengan mengolah data GPS berupa koordinat *latitude* dan *longitude* dalam halaman *Graph User Interface* yang dibuat. Data GPS yang diperoleh akan diproses ketika *button click Track* atau *TrackBack* ditekan atau diaktifkan. Data GPS yang diterima dapat diolah menjadi garis rute dan dapat disimpan dengan menggunakan menu *Save*. Pengolahan data GPS dapat dilakukan sistem jika modul GPS yang digunakan mendapatkan sinyanya GPS. Data GPS yang proses oleh sistem memiliki rata-rata persentase error sebesar 0.0001258%% dengan akurasi 99.999874% pada koordinat *latitude* dan 0.000037% untuk koordinat *longitude* dengan akurasi 99.99996%, serta rata-rata waktu akuisisi selama 299 detik tergantung pada kondisi lingkungan pengujian.
3. Alat *GPS Back Track* menggunakan *Display LCD 800 x 480 Touch Screen* untuk menampilkan proses dan hasil rekaman dengan menggunakan komunikasi HDMI. Program yang telah dibuat dapat menampilkan hasil rekaman dengan membuka file hasil rekaman. Tampilan dari sistem berupa proses dan hasil rekaman yang berlangsung dalam bentuk garis rute perjalanan yang dilakukan.
4. Alat *GPS Back Track* ini menggunakan Kalman filter sebagai filter data koordinat GPS yang digunakan sebagai input pada sistem. Nilai koordinat *latitude* dan *longitude* diproses oleh kalman filter melalui dua tahap. Tahap pertama dalam kalman filter adalah *time update* dimana data *latitude* dan *longitude* diolah untuk mendapatkan nilai pra-estimasi yang digunakan sebagai umpan balik pada proses *measurement update*. Pada proses *measurement update* data akan diolah untuk mendapatkan nilai pasca-estimasi yang hasilnya akan proses kembali untuk dilakukan *Ploting* dan disimpan oleh sistem. Dari hasil pengujian kalman filter diperoleh hasil rata-rata selisih hasil *tracking* pada koordinat *latitude* sebesar 0.000010 dan koordinat *longitude* sebesar 0.000010 yang didapat dari 10 data sampel korrdinat *tracking* pada saat pengujian perekaman rute. kemudian hasil *tracking* rute perjalanan diperoleh hasil perbandingan bentuk grafik dengan

garis yang lebih halus jika dibandingkan garis rute yang terbentuk tanpa menggunakan Kalman filter.

7.2 Saran

Berikut merupakan saran yang digunakan untuk pengembangan dan penelitian lebih lanjut mengenai alat GPS *Back Track* untuk rekaman rute pendakian:

1. Alat GPS *Back Track* pada rekaman rute pendakian ini sebaiknya dibuat dengan menggunakan bahan yang lebih kuat dan dibuat dalam ukuran yang lebih kecil lagi untuk mendapatkan ukuran yang lebih efisien sehingga mudah untuk dibawa.
2. Alat GPS *Back Track* ini menggunakan modul GPS yang hanya mampu bekerja dengan baik jika berada di lingkungan yang terbuka. Maka dari itu, lebih baik menggunakan modul GPS yang lebih baik lagi dalam pencarian sinyal, sehingga proses pengguna dapat dengan mudah mendapatkan sinyal dimanapun posisinya meskipun banyak halangan berupa gedung tinggi atau tebing yang tinggi.



DAFTAR PUSTAKA

- Amazon, 2018. *DIYmall NEO-6M GPS Module*. [Online] Available at: <https://www.amazon.com/DIYmall-AeroQuad-Antenna-Arduino-Aircraft/dp/B01H5FNA4K>
- Cai, X., 2005. On The Performance of the Python Programming Language for Serial and Paralel Scientific Computastion. *Scientific Programming 13*, Volume 31-56.
- software, 2016. *Raspberry Pi3*. [Online] Available at: [https://www.cnx-software.com/wp-content/uploads/2016/02/Raspberry Pi 3 Large.jpg](https://www.cnx-software.com/wp-content/uploads/2016/02/Raspberry_Pi_3_Large.jpg)
- Comunity, M., 2017. *Gunung Merbabu Jalur Selo*. [Online] Available at: http://www.merbabu.com/gunung/gunung_merbabu_selo.php [Diakses 2018].
- Damani, A., 2015. Global Positioning System For Object Tracking. *International Journal of Computer Aplication*, Volume 109, p. 40.
- Display, 5. H., t.thn. *5inc HDMI Display*. s.l.: 5inc HDMI Display.
- Dwiyaniti, M., 2011. Aplikasi GPS Berbasis GSM Modem pada Monitoring Bus. *Jurnal Ilmiah Elite Elektro*, Volume 122-128, p. 1.
- Foundation, P. S., 2018. *The Python Standard Library*. [Online] Available at: <https://docs.python.org/3/library/> [Diakses 18 Juli 2018].
- Hanifa, I., 2016. Uji GPS Tracking Dalam Skala Transportasi Antar Kota. *IJEIS*, Volume 6, p. 175.
- Isrianto, K., 2016. Penerapan Aljabar Vektor Pada GPS.
- Loebis, D., 2004. Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. *Control engineering practice*, 12(12).
- Lukito, A. A., 2015. Rancang Bangun Aplikasi Penanda Lokasi Peta Digital. *Dokumen Karya Ilmiah*.
- Mandalamaya, 2015. *Pengertian dan Cara kerja GPS*. [Online] Available at: <http://www.mandalamaya.com/pengertian-gps-cara-kerja-gps-dan-fungsi-gps/> [Diakses 11 Januari 2018].
- Manual, U., 2017. *5inch HDMI Display*. s.l.: s.n.
- Mazidi, M. A., 2011. The 8051 Microcontroller and Embedded System. Dalam: Hanel, penyunt. *LCD and Keyboard Interfacing*. Taiwan: Dept. of Computer Science and Information Engineering.

- Muryono, T., 2010. *Pemodelan Navigasi Darat Gunung Merapi*. QA75 *Electronic computers. Computer science*.
- Nasa, 2014. *Global Positioning System*. [Online] Available at: <https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS.html> [Diakses July 2018].
- Pradhana, A., 2018. *Mendaki Gunung Kebutuhan atau Sekadar Trend*. [Online] Available at: <https://nalarpolitik.com/mendaki-gunung-kebutuhan-atau-sekadar-trend/> [Diakses 3 November 2018].
- Prihardana, A., 2013. *Pengenalan Navigasi Darat*. [Online] Available at: <http://tngciremai.com/2013/07/pengenalan-navigasi-darat/> [Diakses 2018].
- RaspberryPi, t.thn. *Raspberry Pi Model B*. s.l.: Raspberry Pi Foundation.
- Rimadhani, M. ', 2015. *Pembuatan Peta Navigasi Sarana Umum Kota Semarang Sebagai Data Peta GPS Garmin Map*. 4(ISSN : 2337 - 845X).
- Silitonga, T., 2017. *Pengenalan Navigasi Darat*. [Online] Available at: <http://scdc.binus.ac.id/swanarapala/2017/04/pengenalan-navigasi-darat/> [Diakses 10 Januari 2018].
- Teow, J., 2017. *www.medium.com*. [Online] Available at: <https://medium.com/@jaems33/understanding-kalman-filters-with-python-2310e87b8f48> [Diakses 8 Oktober 2018].
- U-blox, t.thn. *u-blox 6 GPS Modules Data Sheet*. s.l.: U-blox.
- Utomo, V., 2016. *Pengenalan Raspberry Pi*. [Online] Available at: <https://www.slideshare.net/ValerianPurnomo/pengenalan-raspberry-pi-70549602> [Diakses 24 Maret 2018].
- Welch, G., 2006. *An Introduction to the Kalman Filter*. TR 95-041.
- Wicaksono, D. A., 2009. *Rancang Bangun Sistem Navigasi GPS/INS dan Kompas Digital Dengan Kalman Filter Pada Mikrokontroler AVR*, Depok: <https://anzdoc.com/ins-dan-kompas-digital-dengan-kalman-filter-pada-mikrontrole.html>.