

IMPLEMENTASI METODE *STORE AND FORWARD* PADA SISTEM PEMANTAUAN PENDAKI GUNUNG

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Fina Af Idatul Husna
NIM: 125150200111072



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2016

PENGESAHAN

IMPLEMENTASI *STORE AND FORWARD* PADA SISTEM PEMANTAUAN PENDAKI
GUNUNG

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Fina Af Idatul Husna

NIM: 125150200111072

Skripsi ini telah diuji dan dinyatakan lulus pada
23 Agustus 2016

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

Aswin Suharsono, S.T, M.T

NIK: 201102 840919 1 001

Rakhmadhany Primananda, S.T, M.Kom

NIK : -

Mengetahui

Ketua Jurusan Teknik Informatika

Tri Astoto Kurniawan, S.T, M.T, Ph.D

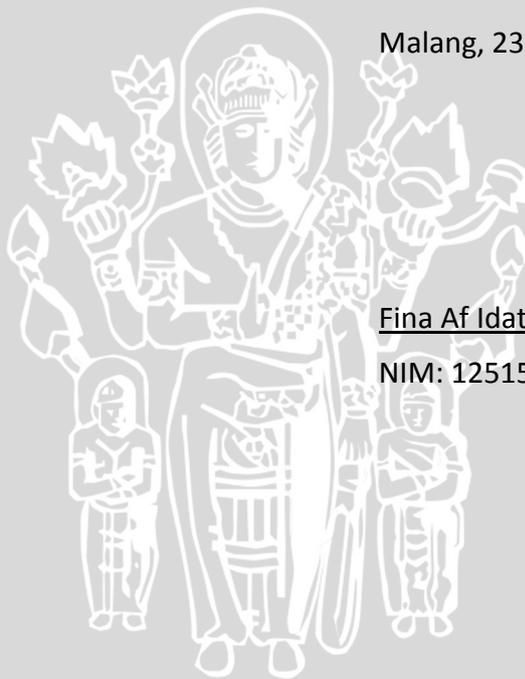
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi 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 disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi 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, 23 Agustus 2016



Fina Af Idatul Husna

NIM: 125150200111072

KATA PENGANTAR

Puji syukur ke hadirat Allah SWT, atas limpahan nikmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi dengan judul “IMPLEMENTASI METODE *STORE AND FORWARD* PADA SISTEM PEMANTAUAN PENDAKI GUNUNG”. Dengan dibangunnya sistem ini diharapkan dapat membantu dalam pemantauan pendaki gunung dan dengan mudah menemukan pendaki yang dinyatakan hilang. Penulis menyusun skripsi ini dengan tujuan memenuhi salah satu syarat untuk memperoleh gelar Sarjana Komputer di Program Studi Teknik Informatika Jurusan Teknik Informatika Fakultas Ilmu Komputer Universitas Brawijaya Malang.

Penulis menyadari bahwa tanpa adanya berbagai pihak yang telah memberikan bimbingan, arahan, dan dukungan maka penulisan skripsi ini tidak akan berjalan lancar. Pada kesempatan kali ini, penulis mengucapkan banyak terima kasih kepada :

1. Aswin Suharsono, S.T., M.T selaku dosen pembimbing 1 yang telah banyak meluangkan waktu untuk memberikan ilmu dan masukan kepada penulis terkait dengan pengerjaan skripsi ini.
2. Rakhmadhany Primananda, S.T, M.Kom selaku dosen pembimbing 2 yang telah banyak meluangkan waktu untuk memberikan ilmu dan masukan kepada penulis terkait dengan pengerjaan skripsi ini.
3. Ungkapan terima kasih spesial untuk kedua orang tua penulis, Ayahanda H. Pudjiono Muhammad Fauzi dan ibunda Sri Wahyuningsih yang selalu senantiasa memberikan doa dan dukungan sehingga penulis dapat menyelesaikan skripsi ini.
4. Bapak Lubis Ibroril Chosam yang telah memberikan banyak informasi tentang studi kasus yang diambil oleh penulis terkait dengan skripsi ini.
5. Seluruh dosen yang telah memberikan ilmu pengetahuan serta motivasi selama penulis menjalani perkuliahan.
6. Teman – teman seperjuangan dalam pengerjaan skripsi keminatan Jaringan Komputer, khususnya Rembulan Suci Fii Jannatin dan Okky Eldiana Muliaputri, yang selalu senantiasa saling memberikan bantuan dan motivasi selama pengerjaan skripsi ini.
7. Romdan Muhammad Ubaidillah dan Arya Yoga Adearta bersedia membantu melengkapi kebutuhan penulis terkait dengan skripsi ini serta saling memberikan motivasi selama pengerjaan.
8. Teman – teman penulis Dewi Maida Safitri, Az Zahra Rahma P.A, Fitri Bibi Suryani, dan Arinie Khaqqo yang selalu memberikan banyak dukungan motivasi dan doa terkait pengerjaan skripsi ini.

9. Teman – teman penulis lain yang tidak dapat disebutkan satu persatu disini atas seluruh motivasi dan bantuan yang telah diberikan.

Penulis berharap semoga seluruh jasa dan kebaikan mereka semua mendapat balasan dari Allah SWT. Penulis menyadari bahwa skripsi yang telah dikerjakan oleh penulis ini masih jauh dari kesempurnaan, maka sangat diharapkan saran dan kritik untuk penelitian selanjutnya dalam pengembangan skripsi ini. Semoga skripsi yang telah dikerjakan oleh penulis ini dapat bermanfaat bagi pembaca terutama teman – teman mahasiswa Fakultas Ilmu Komputer Universitas Brawijaya.

Malang, 23 Agustus 2016

Penulis

finaafidatulhusna@gmail.com



ABSTRAK

Kasus hilangnya pendaki gunung yang semakin bertambah setiap tahunnya telah menjadi topik utama dalam kasus ini. Kasus terakhir yang dapat diambil oleh penulis sebagai acuan adalah pendaki yang hilang di Gunung Semeru yang sampai saat ini masih belum ditemukan. Dari BASARNAS dalam pencarian pendaki gunung yang hilang sejauh ini masih menggunakan beberapa metode yakni pencarian menggunakan *drone* yang dilengkapi *infrared* untuk mengukur suhu badan, menggunakan anjing pelacak serta yang terakhir adalah manual dengan bantuan manusia. Pada penelitian ini penulis mencoba mencari solusi baru untuk membantu tim BASARNAS dalam melakukan pencarian pendaki yang hilang. Solusi yang ditawarkan yakni implementasi metode *store and forward* pada sistem pemantauan pendaki gunung. Daerah pegunungan dikenal dengan daerah yang tidak terdapat jangkauan internet maka dengan metode *store and forward* dapat memudahkan petugas untuk melakukan pemantauan keberadaan pendaki. Metode ini memungkinkan untuk melakukan pengiriman data yakni dengan menggunakan bantuan node perantara yang tidak lain diperankan oleh pendaki itu sendiri memanfaatkan *smartphone* miliknya. Dengan menggunakan metode ini maka petugas akan dengan mudah melakukan pencarian jika mendapati ada pendaki yang hilang. Petugas akan dapat mengetahui keberadaan pendaki terakhir kali sebelum dinyatakan hilang melalui data yang dibawa oleh pendaki. Pengiriman data menggunakan *metode store and forward* akan dapat dengan lancar sampai di tujuan berkat node perantara tanpa harus takut adanya data yang hilang. Sejalan ini pengujian dilakukan pada penelitian mendapatkan hasil sesuai dengan tujuan penggunaan metode. Data yang dititipkan kepada node perantara dapat 100% tersampaikan pada alamat yang dituju tanpa ada data yang hilang.

Kata Kunci : *infrared, store and forward, smartphone*

ABSTRACT

The case of the loss of mountain climbers grow every year has become a major topic in this case. The last case that can be taken by the authors as a reference is a climber missing on the Mount Semeru until now still not found. BASARNAS in search of the missing mountaineers have so far been using several methods, which are finding use *drone* which include infrared to measure body temperature, using sniffer dogs and the latter is a manual with the help of humans. In this research, the authors tried to find new solutions to help the team BASARNAS in searching the missing climbers. The solution offered by the implementation of *store and forward* method in monitoring systems mountaineers. The mountainous area known as the no Internet coverage area with a *store and forward* method can be easier for officers to monitor the presence of climbers. This method makes it possible to perform the data transmission with the help of an intermediary node that no other climber played by itself utilizing his smartphone. Using these methods, the clerk will easily do a search if there find missing climbers. Using these methods, the clerk will easily do a search if there find missing climbers. Officers will be able to know where climbers last time before she went missing over data carried by climbers. Shipping data using *store and forward* method will be able to smoothly reach the destination thanks to an intermediary node without having to be afraid any missing data. So far the testing is done on a research reported results in compatible with the intended use of the methods. The data entrusted to an intermediary node can be 100% passed on to the recipient without any lost data.



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.4.1 Bagi penulis	2
1.4.2 Bagi program studi, fakultas dan universitas.....	2
1.4.3 Bagi para pendaki gunung dan petugas.....	2
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Gunung.....	6
2.2.1 Kasus Pendaki Hilang.....	6
2.3 Metode Stored and Forward	7
2.4 Web Service	9
2.5 JSON	9
2.6 Database MySQL.....	9
2.7 Database Realm	10
BAB 3 METODOLOGI	11
3.1 Studi Literatur	12
3.2 Analisa Kebutuhan Sistem	12



3.3 Analisis Kebutuhan	12
3.4 Perancangan Sistem.....	13
3.5 Implementasi Sistem	15
3.6 Pengujian Sistem.....	15
3.7 Penarikan Kesimpulan	15
BAB 4 PERANCANGAN.....	16
4.1 Arsitektur Sistem.....	17
4.2 Penggunaan Perangkat Lunak Pendukung	18
4.3 Analisis Kebutuhan Aplikasi	19
4.3.1 Kebutuhan Fungsional.....	19
4.3.2 Kebutuhan Non-fungsional	28
4.4 Simulasi Metode <i>Store and forward</i>	29
4.4.2 Mekanisme Pengiriman Data ACK dari Kantor ke Pos	32
4.4.3 Mekanisme Pengiriman Data Titip dari Server Pos ke Server Kantor.....	34
4.5 Perancangan <i>Database</i>	35
4.6 Struktur Pengiriman File JSON	39
4.7 Perancangan Antarmuka Aplikasi Android	41
BAB 5 IMPLEMENTASI	42
5.1 Implementasi Simulasi Jaringan.....	42
5.2 Konfigurasi <i>web server</i> dan alamat ip.....	42
5.3 Konfigurasi menjadikan <i>Raspberry pi</i> sebagai <i>wifi access point</i>	43
5.4 Implementasi <i>Store and forward</i>	45
5.4.1 Implementasi Pendaki memasukkan ID dengan status “naik” ...	46
5.4.2 Implementasi Pendaki memasukkan ID dengan status “turun” .	47
5.4.3 Implementasi Pendaki turun mengambil data titip dari server pos	48
5.4.4 Implementasi Pendaki turun memasukkan data titip ke server kantor	50
5.4.5 Implementasi Pendaki mengupdate status pengiriman di server pos.....	53
5.5 Implementasi Antarmuka	54
5.5.1 Aplikasi Android	54



5.5.2 Tampilan Keberadaan Pendaki 57

BAB 6 PENGUJIAN DAN ANALISIS..... 59

6.1 Pengujian Konektivitas..... 59

6.2 Pengujian Implementasi *Store and forward* 61

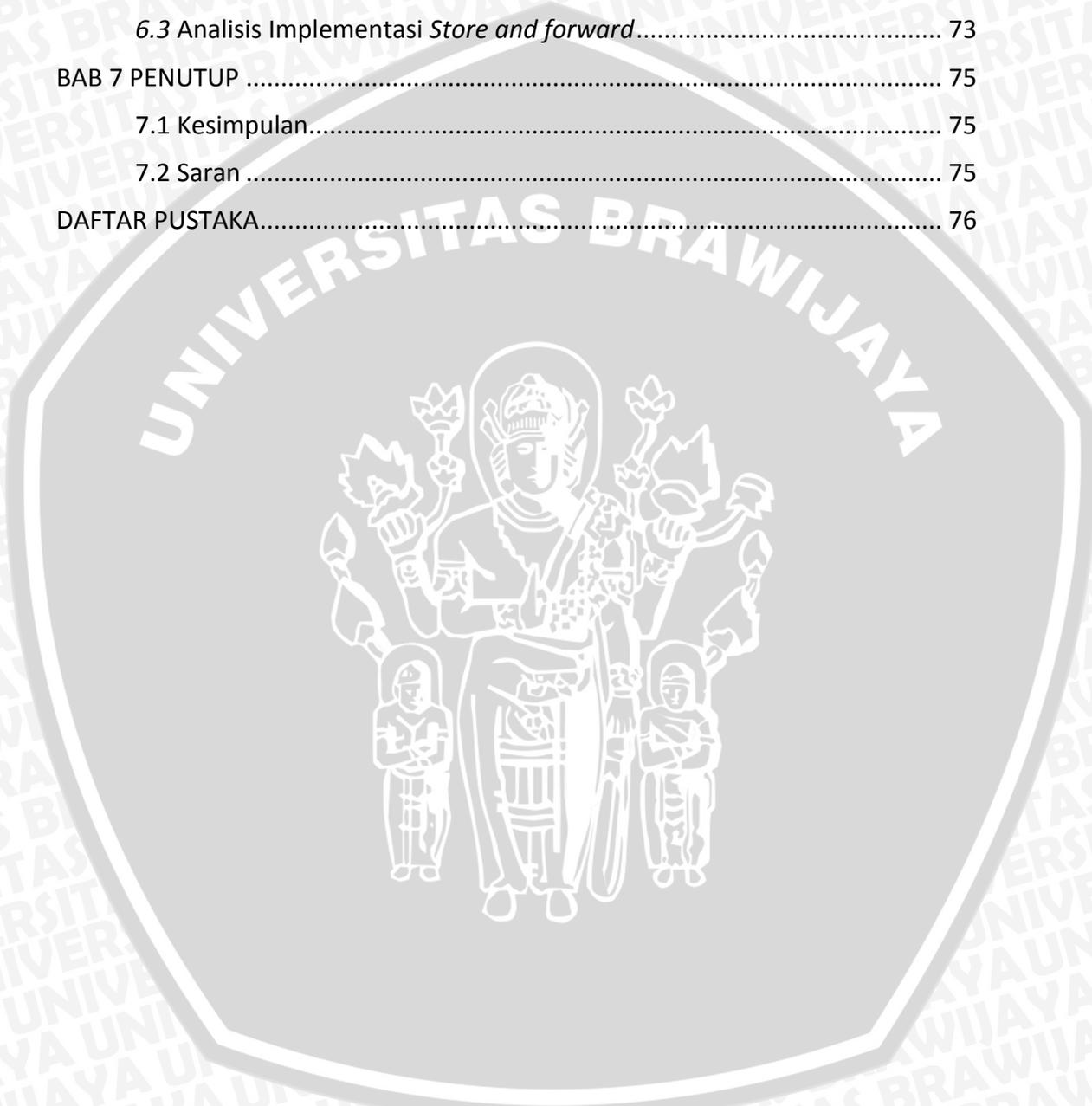
6.3 Analisis Implementasi *Store and forward*..... 73

BAB 7 PENUTUP 75

7.1 Kesimpulan..... 75

7.2 Saran 75

DAFTAR PUSTAKA..... 76



DAFTAR TABEL

Tabel 2.1 Perbandingan dengan Penelitian Sebelumnya	5
Tabel 2.2 Daftar korban pendakian.....	7
Tabel 4.1 Keterangan Status Data pada Server Kantor.....	31
Tabel 4.2 Keterangan Status Data pada Server Pos.....	31
Tabel 4.3 Keterangan Kolom Tabel pos pada serverpos.....	36
Tabel 4.4 Keterangan Kolom Tabel server_pos pada serverpos.....	36
Tabel 4.5 Keterangan Kolom Tabel pos pada serverakhir	38
Tabel 4.6 Keterangan Kolom Tabel server_pos pada serverakhir	38
Tabel 4.7 Struktur Konten Ambil Data Dari Pos.....	39
Tabel 4.8 Struktur Konten Ambil Data Dari Kantor.....	40
Tabel 5.1 Konfigurasi ip static eth0.....	42
Tabel 5.2 Konfigurasi untuk range router.....	43
Tabel 5.3 Konfigurasi ip static wlan0	44
Tabel 5.4 Konfigurasi <i>access point</i>	44
Tabel 5.5 Perintah menjalankan <i>access point</i>	45
Tabel 5.6 Program InsertNaik.php	46
Tabel 5.7 program insertTurun.php.....	47
Tabel 5.8 program ngambilDataTurun.php	49
Tabel 5.9 Program insertserver.php	50
Tabel 5.10 Program ngambilServerBawah.php	52
Tabel 5.11 Program UpdateStatus.php.....	53
Tabel 5.12 Program <i>background service</i> android	55
Tabel 5.13 Program Tampilan <i>Database</i> Kantor.....	57
Tabel 6.1 Pengujian <i>Store and forward</i>	73

DAFTAR GAMBAR

Gambar 2.1 Ilustrasi protokol	8
Gambar 2.2 ilustrasi web services	9
Gambar 3.1 Diagram Alir Metodologi Penelitian.....	11
Gambar 3.2 Alur Kerja Sistem	14
Gambar 4.1 Arsitektur Sistem.....	17
Gambar 4.2 Arsitektur insert ID dan status “naik”	19
Gambar 4.3 Diagram alur insert ID dan status “naik”.....	20
Gambar 4.4 Arsitektur insert ID dan status “turun”	21
Gambar 4.5 Diagram alur insert ID dan status “turun”	22
Gambar 4.6 Arsitektur mengambil data dari pos	23
Gambar 4.7 Diagram alur mengambil data dari pos.....	24
Gambar 4.8 Arsitektur memasukkan data ke kantor.....	25
Gambar 4.9 Diagram alur memasukkan data ke kantor.....	25
Gambar 4.10 Arsitektur mengambil data dari kantor	26
Gambar 4.11 Diagram alur mengambil data dari kantor.....	27
Gambar 4.12 Arsitektur pendaki mengupdate status	27
Gambar 4.13 Diagram alur pendaki mengupdate status.....	28
Gambar 4.14 Simulasi Metode <i>Store and forward</i>	29
Gambar 4.15 Tampilan Pengiriman Data ACK ke Server Pos.....	32
Gambar 4.16 Tampilan Alur Pengiriman ACK ke Server Pos.....	33
Gambar 4.17 Tampilan Pengiriman Data Titip ke Server Kantor.....	34
Gambar 4.18 Tampilan Alur Pengiriman Data Titip ke Server Kantor	35
Gambar 4.19 Rancangan <i>Database</i> Pada Server Pos.....	36
Gambar 4.20 Rancangan <i>Database</i> pada Server Kantor	37
Gambar 4.21 Rancangan <i>database</i> pada aplikasi android.....	39
Gambar 4.22 Rancangan antarmuka aplikasi android	41
Gambar 5.1 Konfigurasi isc-dhscp-server	44
Gambar 5.2 Tampilan utama aplikasi android.....	55
Gambar 6.1 Tampilan <i>access point</i> enable	60
Gambar 6.2 Tampilan pada smartphone	60

Gambar 6.3 Tampilan <i>access point</i> sedang digunakan	61
Gambar 6.4 Tampilan form insert ID pendaki.....	62
Gambar 6.5 Tampilan Pendaki memasukkan id.....	63
Gambar 6.6 Tampilan <i>Background service</i>	64
Gambar 6.7 Tampilan <i>Database</i> Di dalam Server Kantor	65
Gambar 6.8 Tampilan <i>Background service</i> – pos 1	66
Gambar 6.9 Tampilan <i>Database</i> server pos 1.....	66
Gambar 6.10 Tampilan <i>Background service</i> – pos 2	68
Gambar 6.11 Tampilan <i>Database</i> server pos 2.....	68
Gambar 6.12 Tampilan Isi <i>Database</i> Kantor	69
Gambar 6.13 Tampilan <i>background service</i> pendaki “Gina”	70
Gambar 6.14 Tampilan <i>Database</i> Kantor Setelah Diambil Pendaki	71
Gambar 6.15 Tampilan antarmuka berdasarkan pos 1	72
Gambar 6.16 Tampilan antarmuka berdasarkan pos 2	72



BAB 1 PENDAHULUAN

1.1 Latar belakang

Indonesia merupakan negara yang terkenal memiliki berbagai macam wisata alam yang sangat indah. Salah satu wisata alam yang indah di Indonesia yaitu gunung. Banyak gunung di Indonesia yang dijadikan salah satu tujuan utama para wisatawan lokal maupun wisatawan asing saat berkunjung ke Indonesia. Namun, mengingat gunung menawarkan keindahan alamnya selain itu juga dapat menyebabkan musibah bagi para pendakinya.

Dalam periode April 2013 hingga sekarang sudah tercatat 49 orang yang ditemukan telah meninggal dunia ketika melakukan pendakian. Masalah yang diderita yaitu kelelahan saat mendaki, hipotermia, hingga tersesat dan kemudian hilang. Pada 14 Juli 2016 terdapat dua pendaki dilaporkan hilang saat sedang melakukan pendakian di Gunung Slamet (Firdaus, 2016). Kasus lain dalam hilangnya pendaki gunung terjadi pada 7 Juni 2016 yang sedang melakukan pendakian di Gunung Semeru dan hingga saat ini masih belum ditemukan (Suriyanto, 2016).

Dikarenakan tidak adanya jaringan internet di dalam area pegunungan maka dalam pencarian pendaki yang hilang selama ini Tim BASARNAS baru dapat melakukannya dengan cara manual saja yang dapat berlangsung lama serta menghabiskan banyak tenaga. Dewasa ini pencarian dibantu dengan *drone* yang dilengkapi dengan sensor pendeteksi keberadaan manusia, namun pendeteksi ini akan berfungsi jika kondisi pendaki masih dalam keadaan hidup (Chosam, 2015). Dengan bertujuan untuk mempermudah pencarian maka dibutuhkan fungsi dari sistem komputer. Sistem ini dapat digunakan untuk melacak keberadaan terakhir dari pendaki yang diduga hilang dengan begitu maka tim BASARNAS dapat dengan cepat menuju tempat tersebut untuk menemukan korban yang hilang.

Pada penelitian ini maka penulis merancang sebuah sistem pemantauan pendaki gunung dengan menggunakan metode *store and forward*. Di dalam sistem pemantauan ini pendaki harus memiliki ID yang akan diberikan oleh petugas kantor pada saat melakukan pendaftaran dan kemudian ID dimasukkan pada aplikasi pendakian di dalam smartphone milik pendaki. Para pendaki saat melakukan pendakian akan secara otomatis terhubung dengan *access point* yang terdapat di setiap pos dengan memanfaatkan metode *stored and forward*. Kemudian jika ada kasus pendaki yang hilang maka akan dapat diketahui di pos mana dia telah dinyatakan hilang, dan selanjutnya petugas dan tim BASARNAS dapat secara langsung mendatangnya.

Penelitian yang terkait dengan Implementasi pengiriman data dengan menggunakan metode *store and forward* adalah "Autonomous Pickup and Delivery for Delay Tolerant Mobile Networks" (Ozcan Koc, 2005). Dimana pada penelitian ini menggunakan metode *store and forward* pada DTN untuk melakukan pengiriman informasi. Adapun penelitian lain yang menggunakan metode *store and forward* adalah "Pengembangan Sistem Aplikasi Pengiriman

Data Daerah Terpencil Berbasis Delay Tolerant Network” (Siswanti, 2013). Pada penelitian ini membahas bagaimana agar dapat melakukan pengiriman data pada saat PEMILU dilaksanakan di daerah terpencil. Dalam pengiriman data ke kota menggunakan metode *store and forward*.

Berdasarkan permasalahan yang ada terutama pendaki yang hilang maka diusulkan judul “Implementasi Metode *Store and forward* pada Sistem Pemantauan Pendaki Gunung” dengan menggunakan metode *stored and forward* yang digunakan dari sisi pendaki serta petugas. Hasil dari implementasi ini diharapkan dapat bermanfaat dalam melakukan pencarian pendaki yang hilang bagi para petugas serta tim BASARNAS.

1.2 Rumusan masalah

Berdasarkan latar belakang yang telah dijelaskan, maka masalah yang dapat dirumuskan adalah :

1. Bagaimana membangun sistem pemantauan pendaki gunung dengan menggunakan metode *store and forward*?
2. Bagaimana dapat melakukan pengiriman data dengan menggunakan metode *store and forward*?

1.3 Tujuan

Tujuan yang ingin dicapai dari pembuatan skripsi ini adalah:

1. Merancang sistem pemantauan pendaki gunung.
2. Menerapkan metode *stored and forward* dalam sistem pemantauan.
3. Mengetahui hasil uji pemantauan pendaki gunung dengan metode *store and forward*.

1.4 Manfaat

Penulisan tugas akhir ini diharapkan dapat memberikan manfaat serta berguna bagi penulis dan pembaca. Manfaat yang diharapkan adalah:

1.4.1 Bagi penulis

1. Dapat mengetahui pentingnya informasi pendaki dengan menggunakan metode *store and forward*.
2. Dapat menerapkan ilmu yang telah didapatkan ketika perkuliahan terutama materi jaringan komputer.

1.4.2 Bagi program studi, fakultas dan universitas

Sebagai bahan informasi untuk menambah kepustakaan tentang penelitian jaringan sehingga dapat dijadikan referensi pada penelitian selanjutnya.

1.4.3 Bagi para pendaki gunung dan petugas

1. Memberikan alternatif kepada para pendaki gunung yang hilang dengan mengetahui keberadaan pendaki.

2. Mengatasi kesulitan para petugas untuk mencari korban hilang karena keberadaan pendaki yang terlacak secara otomatis.

1.5 Batasan masalah

Batasan masalah yang diterapkan berdasarkan rumusan masalah untuk menghindari semakin melebarnya masalah adalah :

1. Pembahasan difokuskan pada pengiriman data yang bertujuan untuk mengetahui keberadaan pendaki.
2. Pengiriman data tidak bergantung dengan maksimal waktu yang ditentukan.
3. Daya smartphone yang digunakan klien/pendaki.
4. Aplikasi pendakian berbasis *mobile apps android*.
5. Tidak adanya pendaki yang turun untuk membawa data yang akan dititipkan.
6. Pendaki harus melalui puncak untuk dapat terdeteksi dengan status “turun”.

1.6 Sistematika pembahasan

Sistematika pembahasan ditujukan untuk memberikan gambaran dan uraian singkat dari penyusunan tugas akhir yang meliputi beberapa bab, yaitu:

BAB I Pendahuluan

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan skripsi Implementasi Metode *Store and forward* pada Sistem Pemantauan Pendaki Gunung.

BAB II Landasan Kepustakaan

Berisi penjelasan tentang kajian pustaka yang terkait dengan pelaksanaan penelitian dan penerapan yang telah dilakukan sebelumnya terkait dengan metode *store and forward*. Dasar teori yang diperlukan untuk mendukung penelitian ini adalah metode *store and forward*, *raspberry pi* sebagai komponen utama, *TCP/IP*, *web service*, *database MySQL*, *database Realm*, *Android Studio*.

BAB III Metodologi

Pada bab ini membahas tentang metode dan langkah kerja yang akan dilakukan dalam penulisan tugas akhir. Langkah – langkah yang digunakan terdiri dari : studi literatur, analisa kebutuhan, perancangan sistem, implementasi, pengujian sistem dan analisa hasil pengujian, serta penarikan kesimpulan.

BAB IV Perancangan

Pada bab ini membahas tentang bagaimana sistem pemantauan pendaki gunung dapat dilakukan dengan menggunakan metode *store and forward*.

Metode ini akan dilakukan oleh pendaki dengan *Raspberry pi* dengan mengakses *web server* yang telah disediakan.

BAB V Implementasi

Pada bab ini membahas tentang pelaksanaan Implementasi Metode *Store and forward* pada Sistem Pemantauan Pendaki Gunung. Sesuai dengan yang dijabarkan di Bab IV, mengimplementasikan pengiriman data menggunakan metode *store and forward*.

BAB VI Pengujian dan Analisis

Membahas hasil simulasi dengan melakukan pengujian dan menganalisa hasil pengujian yang telah dilakukan sehingga dapat memberikan hasil yang sesuai dengan tujuan penelitian. Pengujian yang dilakukan berdasarkan beberapa skenario yang telah dibuat.

BAB VIII Penutup

Berisi kesimpulan yang didapat dari hasil simulasi “Implementasi Metode *Store and forward* pada Sistem Pemantauan Pendaki Gunung” dan saran yang dapat dikembangkan untuk penelitian lebih lanjut.



BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini akan dibahas mengenai penelitian sebelumnya yang terkait dengan metode *store and forward*, kajian mengenai pengertian metode *store and forward*, kajian mengenai raspberry pi sebagai komponen utama, kajian mengenai TCP/IP, kajian mengenai *web service*, kajian mengenai *database MySQL*, kajian mengenai *database Realm*, dan kajian mengenai *Android Studio*.

2.1 Kajian Pustaka

Pada sub bab kajian pustaka ini akan membahas tentang penelitian terkait yang menerapkan pengiriman pesan menggunakan metode serupa dengan penulis yakni “Autonomous Pickup and Delivery for Delay Tolerant Mobile Networks” (Ozcan Koc, 2005). Pada penelitian ini membahas bagaimana pengiriman pesan dengan mengadopsi metode *store and forward*. Dimana dalam hal ini pengiriman akan dapat dilakukan dengan bantuan node tertentu untuk berperan sebagai *carry*. Adapun skenarionya adalah suatu node tertentu akan mengirimkan pesan kepada node tujuan maka pesan tersebut akan dititipkan melalui node perantara yang berperan sebagai *carry* untuk membantu mengirimkan pesan.

Penelitian lain yang menggunakan metode *store and forward* yakni “Pengembangan Sistem Aplikasi Pengiriman Data Daerah Terpencil Berbasis Delay Tolerant Network” (Siswanti, 2013). Pada penelitian ini membahas bagaimana agar dapat melakukan pengiriman data pada saat pemilu dilaksanakan di daerah terpencil. Adapun skenarionya yakni pengiriman informasi akan dilakukan dari Tempat Pemungutan Suara (TPS) hingga dapat sampai pada Komisi Pemilihan Umum (KPU) Kota dengan bantuan kurir digital.

Tabel 2.1 Perbandingan dengan Penelitian Sebelumnya

No.	Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	Autonomous Pickup and Delivery for Delay Tolerant Mobile Networks (Ozcan Koc, 2005)	Melakukan pengiriman data atau informasi dengan menggunakan metode <i>store and forward</i> .	Pada penelitian ini mengirimkan menggunakan arsitektur DTN yang memiliki metode <i>store and forward</i>	Pada penelitian ini penulis akan mengimplementasikan pengiriman data menggunakan metode <i>store and forward</i> dengan protocol HTTP.
2	Pengembangan Sistem Aplikasi	Melakukan pengiriman data atau	Pada penelitian ini mengirimkan	Pada penelitian ini penulis akan mengimplementasikan

Pengiriman Data Daerah Terpencil Berbasis Delay Tolerant Network (Siswanti, 2013)	informasi dengan menggunakan metode <i>store and forward</i> .	menggunakan arsitektur DTN yang memiliki metode <i>store and forward</i>	pengiriman data menggunakan metode <i>store and forward</i> dengan protocol HTTP.
---	--	--	---

Pada Tabel 2.1 menunjukkan bahwa pada penelitian ini penulis menggunakan dua penelitian sebelumnya sebagai acuan metode. Jika kedua penelitian sebelumnya disebutkan bahwa metode yang digunakan yakni *store and forward* di dalam DTN berbeda dengan penulis yang menggunakan metode *store and forward* dalam komunikasi TCP. Persamaan dari penelitian penulis dengan penelitian sebelumnya yakni membuat solusi untuk pengiriman data di daerah yang kurang akan jaringan internet maka menggunakan metode *store and forward* sebagai solusinya.

2.2 Gunung

Indonesia merupakan negara yang memiliki potensi wisata yang beranekaragam, baik dari wisata alam yakni Puncak gunung hingga dasar laut maupun keragaman budaya yang dimiliki. Gunung merupakan wisata alam yang banyak terdapat di Indonesia. Di dalam kawasan ini sangatlah tidak memungkinkan untuk mendapatkan sinyal atau energi untuk smartphone atau barang elektronik lainnya. Kebanyakan dari para wisatawan lokal maupun asing yang berkunjung ke Indonesia bertujuan untuk menikmati keindahan gunung.

Selain menawarkan banyak keindahan, gunung juga memiliki beberapa faktor yang dapat membahayakan bagi para pendaki. Beberapa faktor itu diantaranya adalah hipotermia, cuaca yang ekstrim, terjatuh ke jurang dan hingga terkena longoran material vulkanik. Salah satu faktor yang sering kali terjadi yaitu hilangnya para pendaki karena tersesat hingga beberapa hari lamanya dan bahkan tidak sedikit diantaranya ditemukan dalam keadaan meninggal dunia.

2.2.1 Kasus Pendaki Hilang

Setiap tahun selalu ada berita tentang hilangnya pendaki yang sedang melakukan pendakian. Dalam hal ini maka berarti bahwa di Indonesia telah membutuhkan banyak cara untuk melakukan pencegahan. Indonesia memiliki BASARNAS untuk melakukan pencarian jika terdapat kasus pendaki hilang. Tim BASARNAS telah melakukan segala upaya agar dapat menyelamatkan pendaki yang hilang salah satunya yakni menggunakan alat terbaru yang disebut *drone*. *Drone* adalah sebuah pesawat terbang tanpa awak yang dijalankan dengan menggunakan *remote control* dimana pada alat ini dilengkapi dengan *infrared* yang dapat mendeteksi suhu tubuh. Maka demikian dengan alat ini akan dapat membantu menemukan pendaki yang hilang dalam keadaan hidup, alat ini tidak akan dapat menemukan pendaki yang sudah dalam keadaan meninggal dunia.



Adapun data pendaki korban pendakian di Pos Ranu semeru sejak 1969 belum termasuk dengan pendaki yang hilang pada 6 Juni 2013 yaitu sesuai dengan Tabel 2.2 :

Tabel 2.2 Daftar korban pendakian

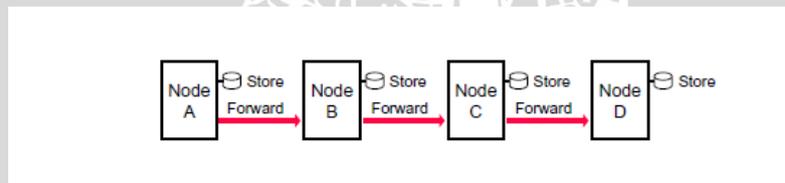
JUMLAH	KETERANGAN
28 Orang	Meninggal Dunia
3 Orang	Tidak ditemukan / meninggal
25 Orang	Mengalami luka-luka / selamat

Sumber : (Istiwawan, 2013)

Dewasa ini telah mendapat berita baru lagi yakni hilangnya dua orang pendaki asal Swiss di Gunung Semeru yang hingga saat ini belum juga ditemukan (Priyasidharta, 2016). Kabar yang diterima dari kasus ini tim BASARNAS tidak dapat memanfaatkan anjing pelacak untuk melakukan pencarian dikarenakan kondisi yang tidak memungkinkan.

2.3 Metode Stored and Forward

Metode stored and forward disini merupakan adopsi dari metode yang dimiliki oleh *Delay Tolerant Network* (DTN). Metode ini menggunakan lapisan baru untuk menyimpan dan meneruskan pesan yang disebut lapisan bundle. Metode ini bekerja dengan meneruskan seluruh pesan atau sebagian dari pesan yang disimpan di sebuah node ditujukan ke node yang lain dengan melalui koneksi jaringan lain (Rahmania, 2013).



Gambar 2.1 Metode Store-and-Forward

Sumber : (Rahmania, 2013)

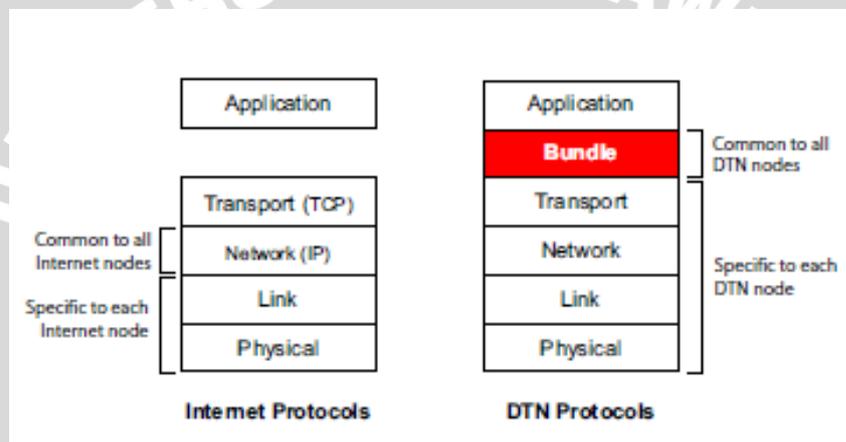
Store and forward merupakan salah satu prinsip yang digunakan dalam arsitektur DTN untuk pengiriman dimana tidak terdapat jalur *end-to-end* (Fall, 2003). Metode *store and forward* sangatlah membantu dalam hal pengiriman data dari node satu ke node yang lainnya. Dalam DTN menggunakan *store and forward* untuk mengatasi masalah yang terkait dengan :

- Konektifitas yang tidak stabil,
- *Variable delay*,
- Tingkat kesalahan yang tinggi (Warthman, 2015).

Metode *store and forward* saat ini telah banyak digunakan dalam sistem voicemail dan email. Namun dalam hal ini tidak dilakukan dari node satu ke node yang lain melainkan dilakukan dari server terpusat kemudian masing – masing sumber menghubungi secara mandiri (Warthman, 2015).

Metode *store and forward* telah menjadi pilihan yang tepat untuk dapat meringankan permasalahan yang ada di daerah – daerah terpencil. Permasalahan yang dapat diselesaikan terutama dalam lingkup pertukaran informasi atau pengiriman data.

Adapun arsitektur DTN untuk mengimplementasikan metode *store and forward* dilakukan pada sebuah transmisi protokol baru yang disebut *bundle protocol*. Jika di dalam protokol internet biasanya dari *transport layer* kemudian dilanjutkan dengan *application layer* maka di dalam protokol DTN terdapat layer tambahan yakni *bundle layer* yang letaknya berada setelah *transport layer*.



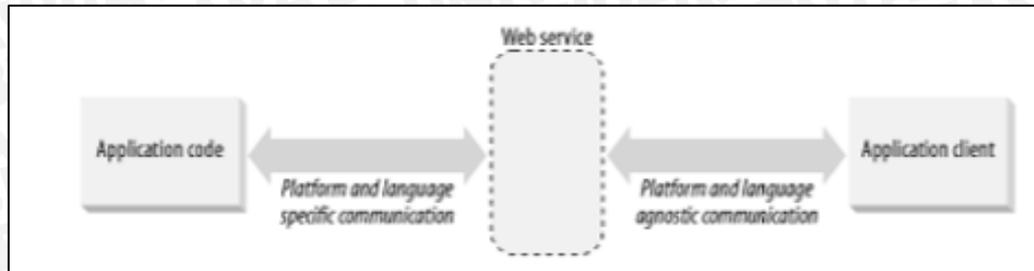
Gambar 2.1 Ilustrasi protokol

(Warthman, 2015)

Pada Gambar 2.1 menunjukkan ilustrasi dari protokol bundle overlay di dalam DTN protocols yang dibandingkan dengan internet protocols.

2.4 Web Service

Web service adalah antarmuka yang dibangun menggunakan standart teknologi internet agar dapat mengakses fungsi aplikasi melalui jaringan (Doug Tidwell, 2001).



Gambar 2.2 ilustrasi web services

(Doug Tidwell, 2001)

Gambar 2.2 menunjukkan bahwa *web service* merupakan sebagai penghubung antara *application code* dengan *application client*. *Client* dapat mengakses atau menjalankan perintah yang ada di dalam *application code* melalui *web service*. Pemanggilan *web service* dapat dilakukan dengan menggunakan platform dan bahasa pemrograman apa saja (Lucky, 2008). *Web service* yang banyak digunakan saat ini adalah HTML. Adapun aplikasi yang menggunakannya yakni diantaranya : aplikasi publikasi, pencarian dan pengambilan konten dimana mengakses *web service* ini melalui protokol standart dengan menggunakan format data HTTP atau HTML. Penggunaan *web service* diterapkan pada bermacam – macam form, dan dapat digunakan dimanapun sesuai dengan keperluan. (Doug Tidwell, 2001).

2.5 JSON

JavaScript Object Notation (JSON) digunakan untuk merepresentasikan tipe data dalam *resource* dengan format berbasis teks (Sari, 2015). JSON dibuat sebagai format teks yang tidak bergantung pada bahasa pemrograman tertentu melainkan menggunakan bahasa pemrograman umum. Dari sifat inilah yang menjadikan format JSON ideal menjadi bahasa pertukaran data (Nazaruddin, 2012). Pengiriman format data dari *web service* dapat berupa JSON dan XML, namun JSON memiliki kelebihan utama dibandingkan XML yakni ukuran file yang lebih kecil (Mitchell, 2013). File dengan ukuran kecil akan berpengaruh untuk aplikasi *mobile* yang membutuhkan respon yang cepat dalam mengakses *web service*. Untuk pengiriman data dalam format JSON dari PHP menggunakan fungsi *json_encode*. Di dalam penelitian ini JSON menggunakan format daftar nilai terurutkan yang dinyatakan sebagai *array*.

2.6 Database MySQL

MySQL merupakan database yang dikenal dengan kecepatan, kekuatan, dan kemudahan dalam penggunaannya sehingga mempermudah pendaki dalam

mengakses, menyimpan, dan memperbarui informasi (David Axmark, 2001). *Database* ini dapat digunakan pada berbagai platform sistem operasi yakni windows, linux, dan sebagainya. Dalam menggunakan *database* ini dapat menggunakan *query standart* yaitu *query SQL* (Structure Query Language). *Query* ini digunakan untuk melakukan komunikasi dengan server basis data relasional.

Perlakuan dalam *database* ini sesuai dengan prinsip *client-server* yakni *client* akan mengirimkan perintah SQL dan kemudian *server* akan mengeksekusi perintah tersebut kemudian mengembalikan lagi kepada *client* (Widhiarti, 2014). *Database MySQL* memiliki dua kategori perintah yakni :

- DDL (Data Definition Language) digunakan melakukan proses definisi data seperti melakukan pembuatan tabel, pembuatan indeks, dan sebagainya.
- DML (Data Manipulation Language) digunakan dalam melakukan manipulasi data seperti mengupdate , menghapus, dan sebagainya.

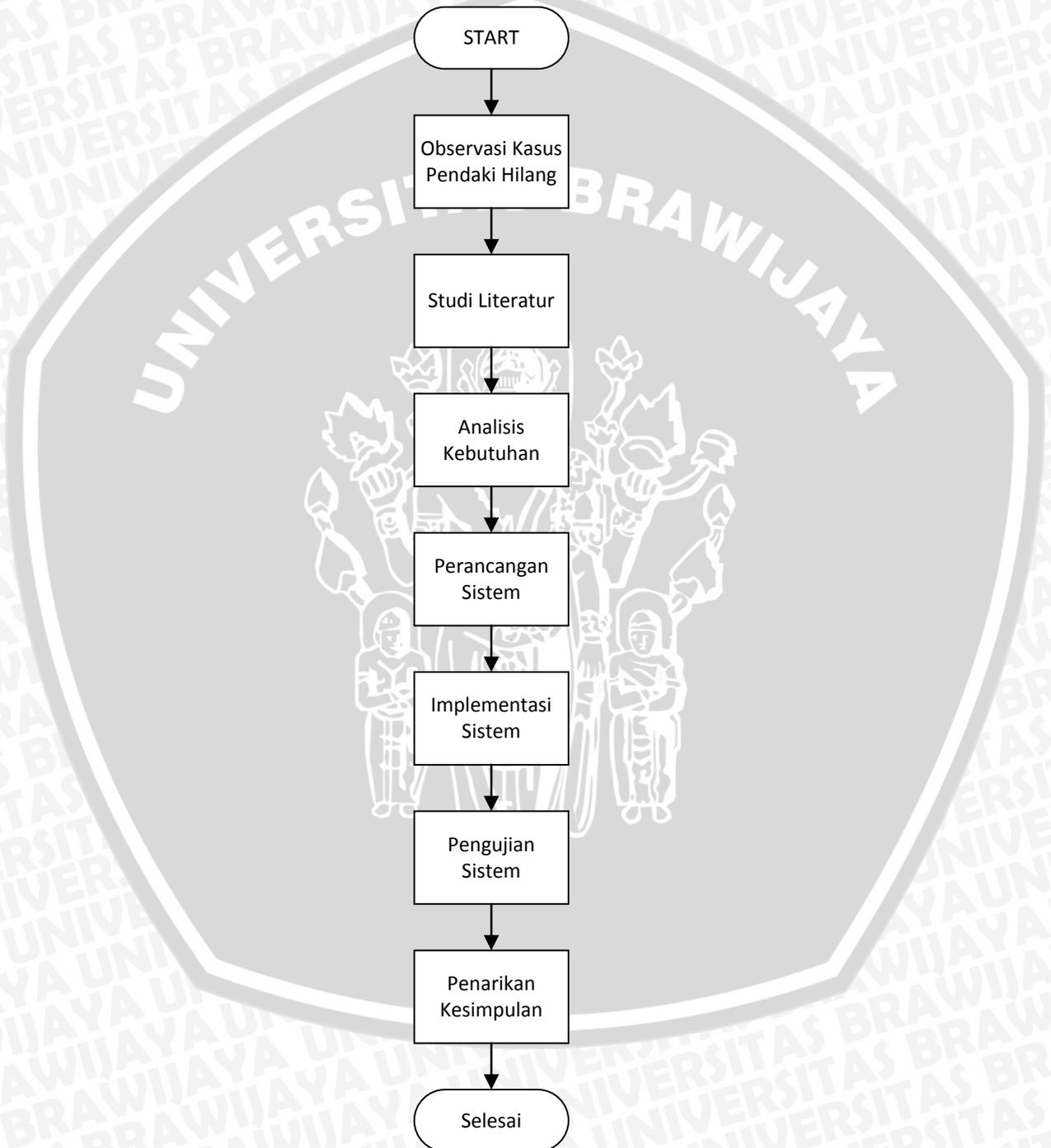
2.7 Database Realm

Database Realm merupakan *database mobile* yang dapat berjalan langsung di dalam ponsel, tablet, ataupun perangkat komunikasi lainnya. *Realm* merupakan *database* yang jauh lebih sederhana daripada *SQLite* dan merupakan pilihan standart setelah belasan tahun terakhir (Koeppel, 2015). *Realm* mendukung platform Android dan iOS serta file dapat digunakan model objek yang sama untuk Swift, Objective-C, Java, React Native, dan Xamarin. *Database realm* tidak dibangun diatas *SQLite* melainkan berdiri sendiri diatas kinerja yang optimal serta memori yang digunakan *smartphone* sangat efisien untuk menggunakannya (Maskov, 2015).



BAB 3 METODOLOGI

Bab ini akan menjelaskan tahapan penelitian yang dilakukan dan penjabaran kebutuhan sistem dalam membangun penelitian implementatif dengan tujuan menyelesaikan permasalahan menggunakan metode *store and forward*. Tahapan metodologi penelitian dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

3.1 Studi Literatur

Studi literatur bertujuan untuk menjadi dasar teori dalam melakukan penelitian ini. Jenis literatur yang dapat digunakan yaitu artikel, buku, jurnal, e-book, website serta laporan penelitian sebelumnya. Di dalam penelitian ini studi literatur yang digunakan meliputi konsep pengiriman data dengan metode *Store and forward*, penjelasan perangkat lunak android sebagai aplikasi pendaki atau node yang juga berperan sebagai perantara di dalam pengiriman data, serta konsep untuk mengimplementasikan *access point* di dalam *Raspberry pi* dengan tujuan menjadi sebagai jaringan perantara antara aplikasi android dengan *web server* yang ada di dalam *Raspberry pi*.

3.2 Analisa Kebutuhan Sistem

Pada sub bab ini akan menganalisis kebutuhan yang diperlukan oleh penulis dalam melakukan implementatif sistem. Kebutuhan sistem disesuaikan dengan kebutuhan penelitian secara keseluruhan, yakni meliputi :

1. Kebutuhan Hardware meliputi :
 - *Personal Computer (PC) / Laptop / Notebook*
 - *Smartphone Android versi 4.2.2*
 - *Raspberry pi 3 model B*
 - USB Adapter TL-WN722N – TP-Link
 - Memory card microSD 4Gb
2. Kebutuhan Software meliputi :
 - Sistem operasi Raspbian
 - Android Studio untuk pembuatan aplikasi android
 - Genymotion untuk demo aplikasi android
 - *Database MySql* untuk di setiap *Raspberry pi*
 - *Database Realm* untuk di aplikasi android

3.3 Analisis Kebutuhan

Pada sub bab menentukan apa saja yang mampu dilakukan oleh sistem dan aplikasi android yang akan dirancang. Sistem yang dimaksud adalah sistem yang mampu mewujudkan mekanisme bagaimana pendaki dapat mengirimkan data dan menerima data melalui aplikasi android pada daerah pegunungan dengan menggunakan jaringan lokal yang didapat dari raspberry di setiap pos pegunungan. Adapun kebutuhan fungsional sistem antara lain yaitu :

- Memiliki aplikasi android untuk menginputkan ID yang akan menjadi pengenalan tiap pendaki.

- Memiliki *web server* dan *web service* pada *Raspberry pi* yang dapat diakses oleh aplikasi pendaki.
- Memiliki *database* di *Raspberry pi* untuk menyimpan data yang telah dikirimkan.
- Memiliki *database* di aplikasi android untuk menyimpan data yang akan dikirimkan.
- Memiliki tampilan monitoring keberadaan pendaki pada kantor utama petugas.

Selain kebutuhan fungsional diatas, sistem juga memiliki kebutuhan non fungsional yakni :

- *Raspberry pi* berperan sebagai *access point* yang menjadi penghubung antara *web server* dengan aplikasi android melalui jaringan wifi.
- Pendaki cukup menyambungkan smartphone dengan wifi lokal yang telah disediakan.
- Aplikasi android ini merupakan aplikasi ringan yang akan mudah digunakan oleh tiap pendaki.
- Pendaki cukup menginputkan ID satu kali di pos pendaftaran, selanjutnya aplikasi akan berjalan dengan *background service*.

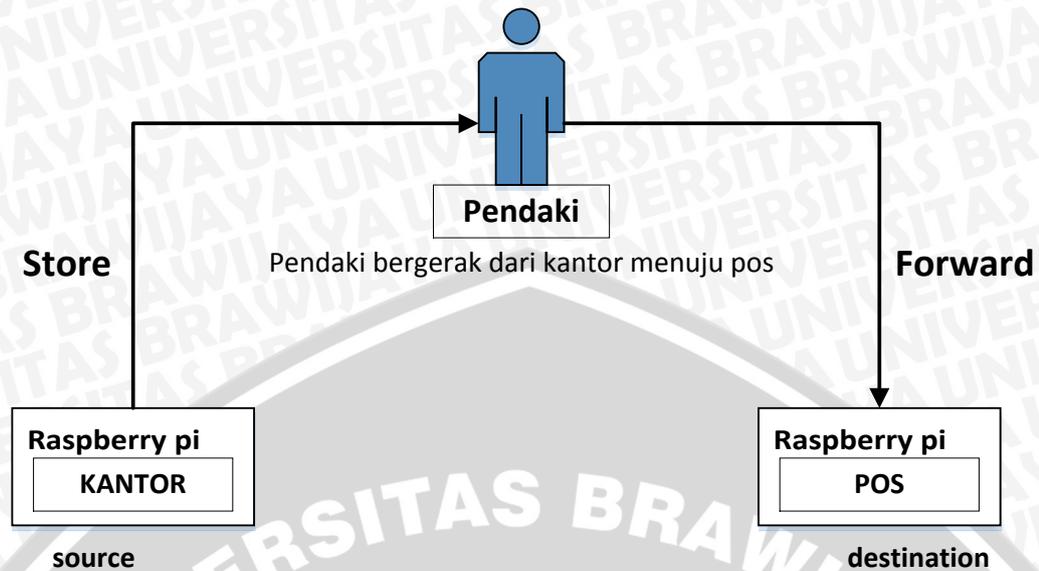
3.4 Perancangan Sistem

Perancangan sistem yang dibangun berdasarkan dari hasil analisis kebutuhan dengan menggambarkan arsitektur sistem sesuai dengan kebutuhan fungsional dan non fungsional yang ditentukan. Serta teori yang telah diperoleh pada kajian pustaka akan diimplementasikan dalam pembuatan sistem pemantauan pendaki gunung dengan proses pengiriman menggunakan metode *store and forward*.

1. Alur kerja sistem

Dalam tahap ini akan digambarkan perancangan sistem yang akan menyelesaikan permasalahan. Alur jalannya sistem secara umum dapat dilihat pada Gambar 3.2.





Gambar 3.2 Alur Kerja Sistem

Pada Gambar 3.2 menunjukkan mekanisme alur kerja sistem dimana komunikasi hanya dapat dilakukan antara server pos dan server kantor dengan pendaki. Dalam sistem ini untuk server pos dengan server kantor tidak dapat melakukan komunikasi secara langsung. Adapun untuk pengiriman data dapat dilakukan dengan melewati kurir yakni diperankan oleh pendaki. Pendaki akan membantu kantor dan pos dalam melakukan pengiriman data, dengan cara data yang akan dikirimkan akan dititipkan kepada pendaki untuk dikirimkan kepada tujuan. Contoh seperti pada Gambar 3.2 dimana server pos akan mengirimkan data kepada destination yang diinginkan yakni server kantor, maka server pos akan terlebih dahulu menitipkan kepada pendaki dan kemudian pendaki akan mengirimkan kepada server kantor.

2. Perancangan Antarmuka

Perancangan antarmuka bertujuan untuk mempermudah mengimplementasikan sistem yang dibuat.

3. Perancangan Pengujian

Perancangan pengujian sistem meliputi :

- Pengujian dalam memasukkan ID dengan status “naik” melalui android ke dalam *database* tiap pos.
- Pengujian dalam memasukkan ID dengan status “turun” melalui android ke dalam *database* tiap pos.
- Pengujian dalam android mengambil data dari pos dalam *background service*.
- Pengujian dalam memasukkan data melalui android ke dalam *database* kantor.

- Pengujian dalam mengirim ack dari kantor ke android untuk dibawa ke tiap pos.
- Pengujian dalam mengupdate status di setiap pos menyesuaikan dengan data yang dibawa dari kantor.

3.5 Implementasi Sistem

Implementasi sistem merupakan tahap dimana sistem dibangun sesuai dengan perancangan yang telah dibuat dan menerapkan teori yang ada dalam studi literatur. Dalam hal ini penulis akan membuat sebuah aplikasi android yang berperan sebagai perantara dalam pengiriman data. Penulis akan membangun satu *web server* beserta *web service* di dalam sebuah perangkat berbasis linux untuk menjadi server kantor dilengkapi dengan antarmuka pemantauan lokasi pendaki berbasis web, kemudian penulis juga akan membangun server di dalam beberapa *Raspberry pi* beserta *access point* untuk sejumlah pos yang ada di pegunungan. Selanjutnya dari perangkat yang telah dibangun akan bertujuan untuk mengetahui keberadaan pendaki yang telah mempunyai ID masing-masing.

3.6 Pengujian Sistem

Proses pengujian bertujuan untuk mengetahui tingkat keberhasilan pembuatan sistem sesuai dengan kebutuhan fungsional dan non fungsional yang telah ditentukan. Dalam hal ini termasuk melakukan pengujian terhadap pengiriman data dengan metode *store and forward*. Selain itu pengujian ini akan dilakukan sesuai dengan skenario pengujian yang telah dibuat. Dalam pengujian ini akan menggunakan aplikasi android untuk melakukan tahap pengujian sesuai yang telah dituliskan di perancangan sistem.

3.7 Penarikan Kesimpulan

Pada sub bab ini akan dapat dilakukan setelah penulis telah menyelesaikan semua tahap termasuk di dalamnya : perancangan sistem, implementasi sistem, serta pengujian sistem. Dalam hal ini penulis akan dapat melakukan pengambilan kesimpulan dari semua tahap yang telah dilalui dan sistem yang telah dibangun. Kesimpulan disini merupakan suatu jawaban dari semua rumusan masalah yang telah ditulis. Selain itu penulis akan menuliskan saran yang akan digunakan untuk pertimbangan dalam penelitian selanjutnya.

BAB 4 PERANCANGAN

Pada bab ini akan menjelaskan perancangan sistem untuk menyelesaikan masalah sesuai dengan latar belakang dan rumusan masalah. Sesuai dengan masalah yang diangkat maka studi kasus dalam penelitian ini yakni dengan tujuan memantau keberadaan pendaki gunung untuk mengantisipasi adanya pendaki yang hilang.

Dapat dilihat bahwa studi kasus yang diangkat berada di wilayah jauh dari jaringan internet yakni pegunungan. Maka jika terjadi adanya pendaki yang hilang akan sangat kesulitan karena jauh dari jangkauan seluler ataupun internet. Maka penulis mencoba untuk ikut serta membantu dalam pencarian pendaki yang hilang dengan mengimplementasikan metode *store and forward* pada sistem pemantauan pendaki gunung.

Hal pertama yang dilakukan penulis adalah membangun *web server* dan *web service* pada *Raspberry pi* untuk selanjutnya diletakkan pada tiap pos yang berada di pegunungan. Penulis membangun *web server* dengan tujuan menjadi server berisi perintah yang akan diakses oleh aplikasi pendaki sedangkan *web service* merupakan sebagai penghubung aplikasi pendaki dengan perintah yang ada di dalam *web server*. *Raspberry pi* juga dijadikan sebagai *access point* sebagai jaringan wifi lokal untuk dapat diakses oleh pendaki. Para pendaki yang telah melakukan pendaftaran maka akan mendapatkan petunjuk dari petugas untuk memasang aplikasi android yang disediakan untuk pemantauan serta menghubungkan wifi pada smartphonenya dengan wifi *Raspberry pi* yang telah disediakan.

Di dalam *web server* terdapat perintah apa saja yang akan dilakukan oleh aplikasi android. Perintah tersebut diantaranya adalah pendaki diminta untuk memasukkan ID dengan kemudian ditujukan ke dalam *database* yang disediakan, pendaki mengambil data yang telah disediakan oleh tiap pos untuk dibawa turun ke server kantor, pendaki mengupdate status data di dalam *database* sebagai tanda status pengambilan, pendaki memasukkan data yang telah dibawa ke *database* di server kantor yang kemudian dapat ditampilkan, serta pendaki dapat mengupdate status pengiriman pada setiap pos dengan menggunakan *background service*.

Selanjutnya penulis akan melanjutkan dengan pembuatan aplikasi android dengan tampilan sederhana serta dilengkapi dengan *background service* dimana aplikasi akan berjalan tanpa harus merepotkan pendaki. Aplikasi akan secara *background service* menjalankan perintah di dalam *web server*. Aplikasi ini akan secara langsung mengakses *web server* saat dijalankan dan telah terhubung pada wifi yang sama yakni tetap tersambung dengan *access point* yang telah disediakan. Aplikasi android yang dibuat ini ditujukan untuk pendaki dan berfungsi sebagai perantara dalam pengiriman data. Para pendaki hanya akan memasukkan ID satu kali di awal aplikasi dijalankan dan kemudian pendaki dapat menyimpan *smartphone* dengan aman serta melanjutkan perjalanan dengan

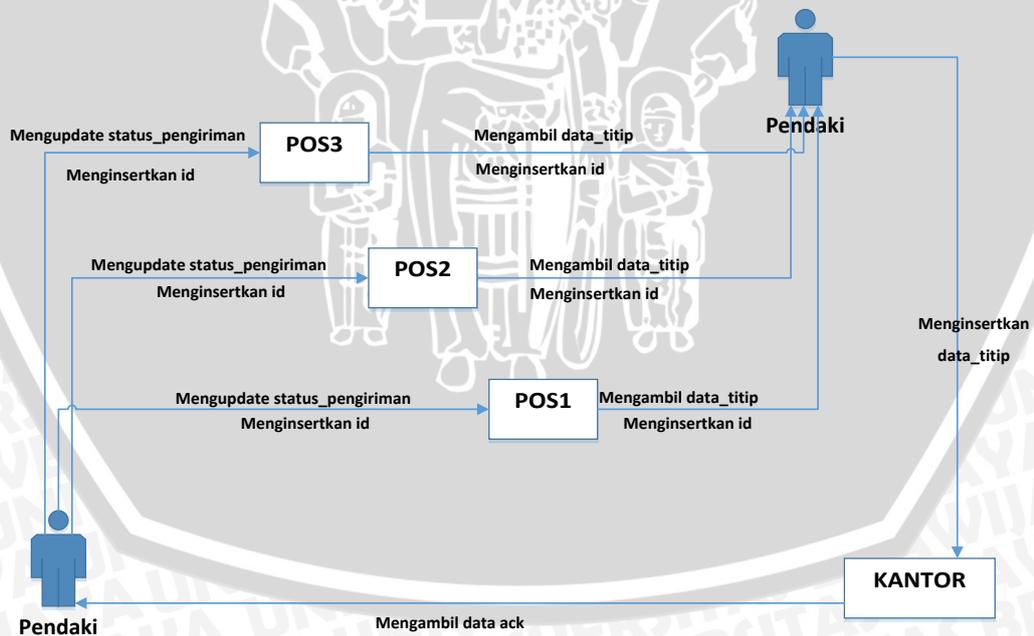
nyaman. Pendaki tidak akan diminta untuk melakukan sesuatu pada aplikasi lagi setelah memasukkan ID miliknya karena aplikasi telah berjalan secara background service. Setiap pendaki akan mendapatkan ID dari petugas ketika melakukan pendaftaran di kantor. ID pendaki ini akan dapat berupa data unik yang akan diberikan oleh petugas dan setiap pendaki tidak akan memiliki ID yang sama.

Pada dasarnya pegunungan merupakan daerah dimana tidak terdapat koneksi internet sehingga jika terjadi hilangnya pendaki akan dilakukan pencarian secara manual. Namun dengan adanya alat ini maka penulis berniat untuk membantu tim BASARNAS untuk melakukan pencarian dengan lebih awal dapat mengetahui lokasi terakhir keberadaan pendaki.

Dalam penelitian ini pertukaran data akan dilakukan di dalam area access point yang diperankan oleh *raspberry pi* pada setiap pos pendakian. Adapun *raspberry* berperan sebagai *web server* yang akan diakses oleh pendaki dengan menggunakan aplikasi androidnya. Ketika pendaki berhasil mengakses *web server* maka akan secara otomatis aplikasi akan menyesuaikan perintah apa saja yang akan dilakukan dengan dijalankan melalui *background service*.

4.1 Arsitektur Sistem

Sesuai dengan penjabaran sistem diatas, maka pada Gambar 4.1 akan mengilustrasikan jalannya sistem menggunakan metode *store and forward* dalam pengiriman data.



Gambar 4.1 Arsitektur Sistem

Sistem yang dibangun oleh penulis adalah suatu sistem pengiriman data dengan menggunakan metode *store and forward*. Menggunakan metode ini

bertujuan untuk tetap dapat memantau pendaki untuk mengantisipasi adanya pendaki yang hilang. Sesuai dengan yang telah digambarkan terdapat pendaki yang baru datang dan menghampiri kantor untuk melakukan pendaftaran serta diarahkan untuk menjalankan aplikasi android pemantau di dalam smartphone miliknya serta menyambungkan wifi lokal yang disediakan.

Setelah pendaki menjalankan aplikasinya maka akan secara langsung server kantor akan menitipkan data ack untuk disebar di setiap pos yang akan dilaluinya dengan tanpa sepengetahuan pendaki. Data ack adalah data yang bertujuan untuk memberikan informasi bahwa data yang dikirimkan oleh pendaki turun telah masuk ke server kantor. Pendaki dapat secara langsung memasukkan ID yang telah diberikan petugas ke dalam aplikasinya. Dengan begitu pendaki dapat melanjutkan pendakian dengan tenang karena tidak akan diminta untuk memasukkan ID lagi di setiap posnya. Perlakuan akan dilakukan aplikasi secara *background service*, dengan tetap menyambungkan koneksinya ke wifi yang disediakan.

Setelah pendaki melanjutkan perjalanan dan menemukan pos pendakian ditengah perjalanannya maka akan secara langsung aplikasinya memasukkan ID pendaki yang telah diberikan oleh petugas sesuai perintah di dalam *web server*, pendaki akan memasukkan dengan status “naik” hingga mencapai pos puncak pendakian serta akan melakukan update *database* dengan data yang dibawa dari server kantor sesuai dengan perintah yang disediakan *web server*. Jika pendaki sudah sampai pos puncak maka pendaki akan secara langsung memasukkan ID dengan status “turun” serta mengambil data dari tiap pos sesuai dengan perintah *web server* untuk dibawa ke server kantor, untuk melakukan tahap ini pendaki harus melalui pos puncak terlebih dahulu Karena jika tidak maka tidak dapat melakukan tahap ini dan tahap berikutnya. Selanjutnya jika pendaki sudah sampai di server kantor maka akan secara langsung memasukkan data yang dititipkan setiap pos untuk dimasukkan ke dalam *database* server kantor sesuai dengan perintah *web server*.

4.2 Penggunaan Perangkat Lunak Pendukung

Perangkat lunak yang dibutuhkan untuk mewujudkan skenario implementasi metode *store and forward* pada sistem pemantauan pendaki gunung yaitu :

- Operasi Sistem

Pada setiap pos dan kantor akan diletakkan *Raspberry pi* dengan sistem operasi Raspbian. Selanjutnya kantor menggunakan windows untuk mengakses web sebagai antarmuka petugas, sedangkan pendaki membutuhkan sistem operasi Android di dalam smartphone miliknya.

- *Web server*

Perangkat lunak *web server* yang digunakan adalah Apache *Web server* yang akan dipasangkan di setiap pos dan kantor petugas.

- Web Service

Web service dipasang di dalam setiap server yang ada dengan tujuan aplikasi android dapat mengakses *web server* dalam satu jaringan.

- Database

Sistem ini menggunakan dua macam *database* yakni MySQL dan Realm. Dimana *database* MySQL dipasang di dalam setiap servernya untuk penyimpanan data sedangkan Realm merupakan *database* yang digunakan di dalam aplikasi android.

- Aplikasi Android

Android disini digunakan untuk menjadi perantara dalam pengiriman data dan dijalankan pada *smartphone* pendaki.

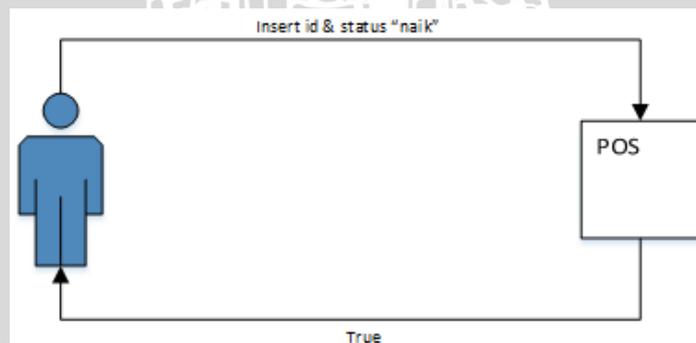
4.3 Analisis Kebutuhan Aplikasi

Pada penelitian ini penulis akan membangun aplikasi android yang berperan sebagai perantara ketika pengiriman data. Data dikirimkan melalui *smartphone* yang telah terhubung dengan *access point* yang sudah tersedia di setiap pos ataupun kantor. Pada pembuatan sistem ini penulis menggunakan bahasa java untuk aplikasi android dan php untuk pembuatan *web server*.

4.3.1 Kebutuhan Fungsional

Pada kebutuhan fungsional disini disesuaikan dengan analisis kebutuhan yang telah dijelaskan sebelumnya serta sesuai dengan batasan masalah yang telah ditentukan. Penjelasan setiap poin dari kebutuhan fungsional adalah sebagai berikut :

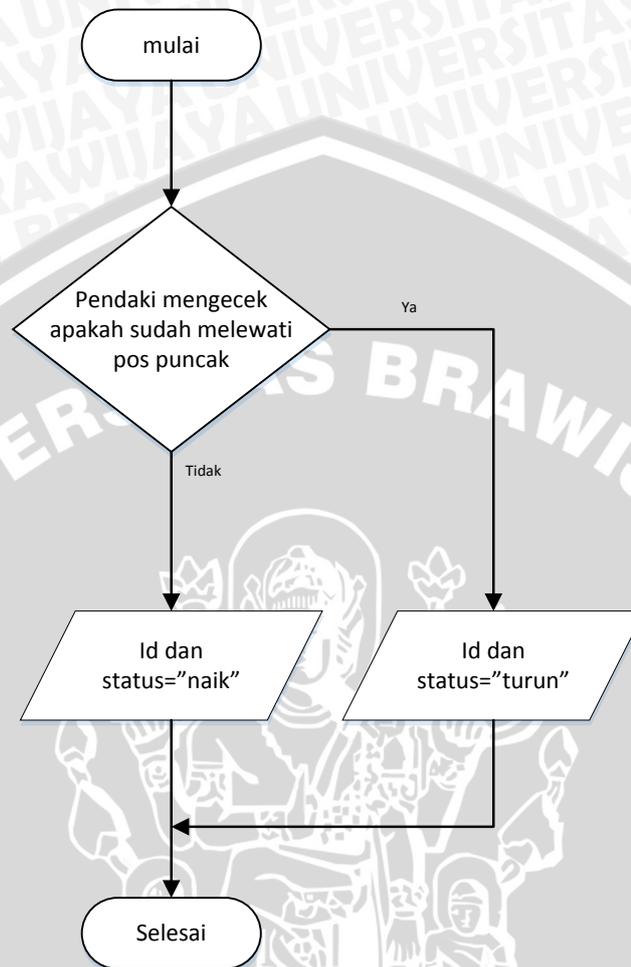
1. Pendaki memasukkan ID dengan status "naik".



Gambar 4.2 Arsitektur insert ID dan status "naik"

Pada tahap ini sesuai dengan Gambar 4.2, pendaki dapat memasukkan ID pada setiap pos dimana skenarionya adalah pendaki yang baru datang akan menjalankan aplikasi android kemudian memasukkan ID dan menekan tombol "kirim". Selanjutnya aplikasi akan secara *background service* mengirimkan ID jika terdapat berada di lingkungan pos. Dengan begitu ID akan secara langsung terdaftar di dalam *database*

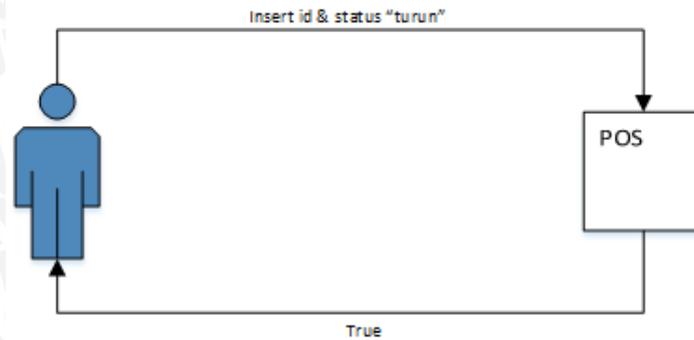
pos dengan status “naik”. Status ini akan menjadi “turun” jika pendaki telah melewati pos puncak. Adapun alurnya dapat dilihat pada Gambar 4.3.



Gambar 4.3 Diagram alur insert ID dan status “naik”

Pada Gambar 4.3 menunjukkan alur dari mekanisme pendaki memasukkan data dengan status “naik” yang dilakukan secara *background service*. Karena berjalan secara *background service* maka proses ini akan terjadi tanpa harus melibatkan pendaki.

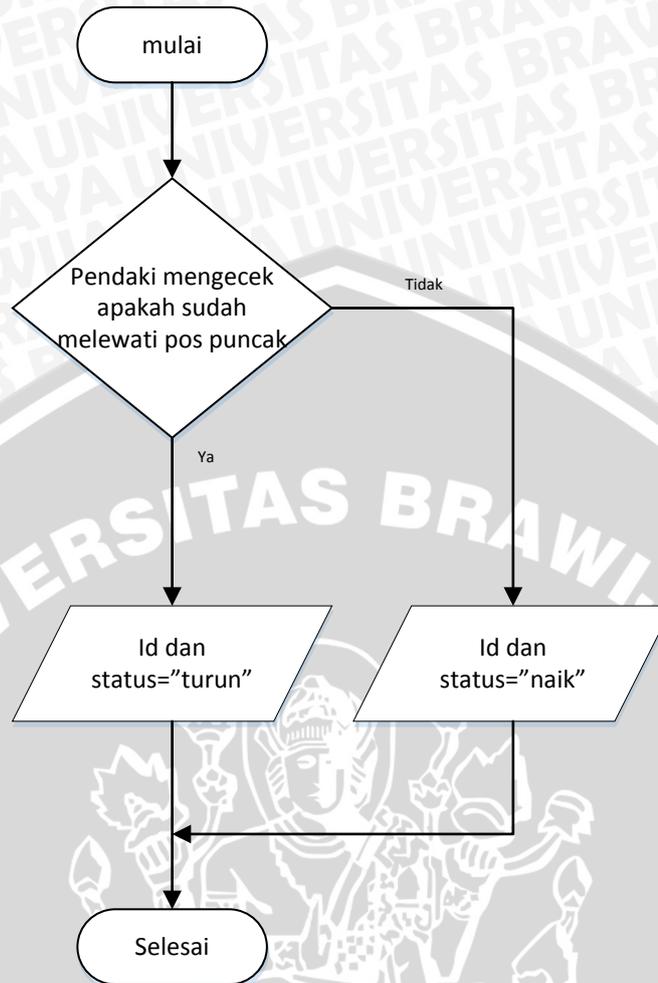
2. Pendaki memasukkan ID dengan status “turun”.



Gambar 4.4 Arsitektur insert ID dan status “turun”

Pada tahap ini sesuai dengan Gambar 4.4, pendaki tidak akan melakukan apapun hanya tetap menyambungkan wifi saja maka aplikasi akan secara langsung memasukkan ID dengan status “turun” setelah melewati pos puncak. Jika pendaki telah melewati pos puncak maka berarti pendaki telah bertujuan untuk turun. Adapun alurnya dapat dilihat pada Gambar 4.5.

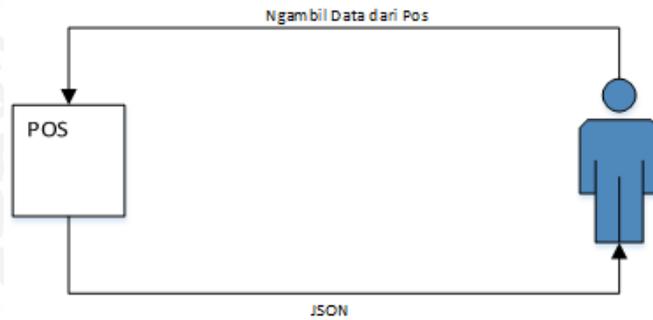




Gambar 4.5 Diagram alur insert ID dan status “turun”

Pada Gambar 4.5 menunjukkan alur dari mekanisme pendaki memasukkan data dengan status “turun” yang dilakukan secara *background service*. Karena berjalan secara *background service* maka proses ini akan terjadi tanpa harus melibatkan pendaki.

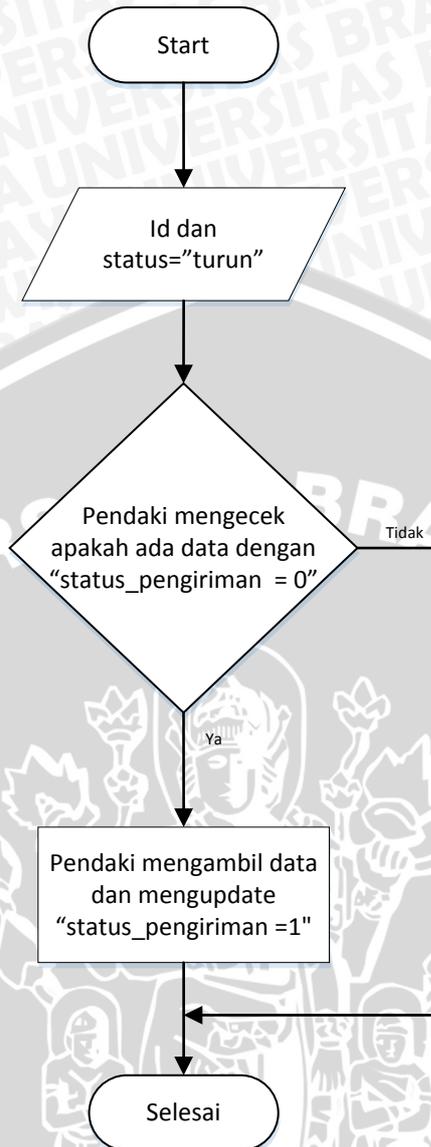
3. Pendaki mengambil data dari setiap pos.



Gambar 4.6 Arsitektur mengambil data dari pos

Pada tahap ini sesuai dengan Gambar 4.6, pendaki turun merupakan pendaki yang telah memasukkan ID dengan status “turun”. Pendaki ini akan secara otomatis mengambil data yang telah disediakan server untuk dititipkan dan dibawa ke server kantor. Data yang akan diambil oleh pendaki adalah data yang mempunyai status pengiriman baru dimana dalam database ditandai dengan data yang memiliki “status_pengiriman = 0”. Jika data sudah dibawa oleh pendaki maka nilai “status_pengiriman” akan berubah menjadi 1 yang berarti sedang dikirim. Adapun alurnya dapat dilihat pada Gambar 4.7.





Gambar 4.7 Diagram alur mengambil data dari pos

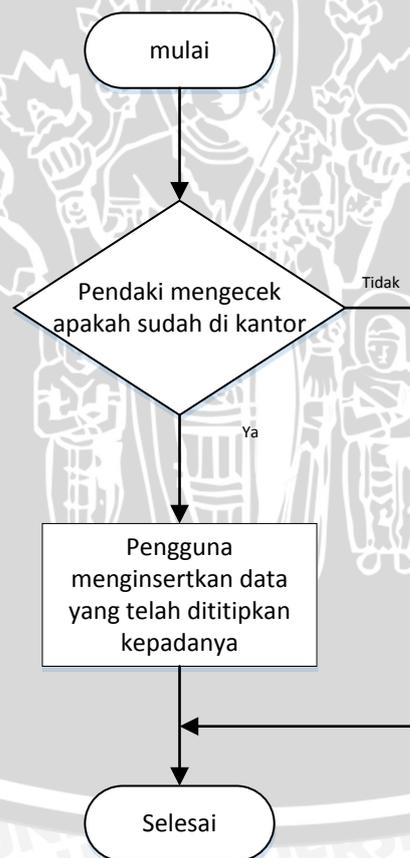
Pada Gambar 4.7 menunjukkan alur dari mekanisme pendaki mengambil data dari server pos yang dilakukan secara *background service*. Karena berjalan secara *background service* maka proses ini akan terjadi tanpa harus melibatkan pendaki.

4. Pendaki memasukkan data titip dari server pos ke dalam server kantor.



Gambar 4.8 Arsitektur memasukkan data ke kantor

Pada tahap ini sesuai dengan Gambar 4.8, pendaki akan memasukkan data yang dititipkan oleh setiap pos ke dalam server kantor. Data yang dibawa oleh pendaki akan otomatis dimasukkan oleh aplikasi tanpa sepengetahuan pendaki melainkan dijalankan oleh *background service*. Adapun alurnya dapat dilihat pada Gambar 4.9.



Gambar 4.9 Diagram alur memasukkan data ke kantor

Pada Gambar 4.9 menunjukkan alur dari mekanisme pendaki memasukkan data yang dibawa dari server pos ke dalam *database* server

kantor secara *background service*. Karena berjalan secara *background service* maka proses ini akan terjadi tanpa harus melibatkan pendaki.

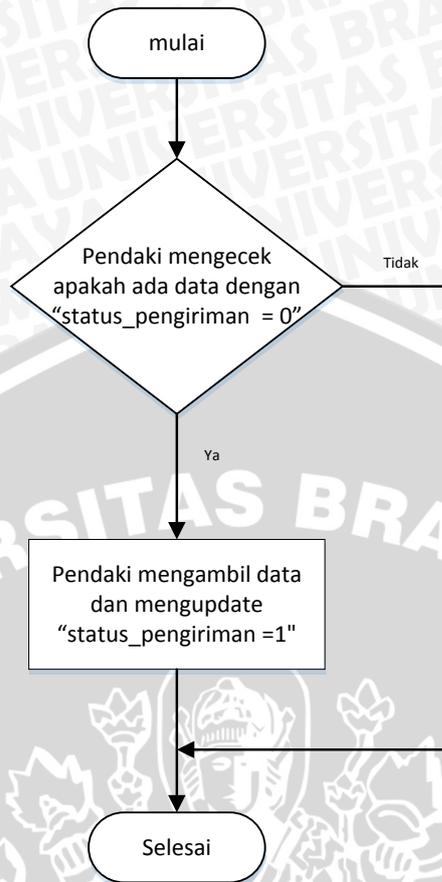
5. Kantor menitipkan ACK kepada pendaki baru.



Gambar 4.10 Arsitektur mengambil data dari kantor

Pada tahap ini sesuai dengan Gambar 4.10, ketika ada pendaki baru yang melakukan pendaftaran di kantor dan telah menyambungkan wifi ke jaringan lokal yang disediakan serta menjalankan aplikasinya maka akan mengambil data yang telah disediakan oleh server untuk dititipkan ke pendaki. Aplikasi akan secara langsung mengambil data tanpa sepengetahuan pendaki. Data yang akan diambil oleh pendaki adalah data yang mempunyai status pengiriman belum terkirim dimana dalam database ditandai dengan data yang memiliki "status_pengiriman = 0". Alur untuk skenario tahap ini dapat dilihat pada Gambar 4.11.

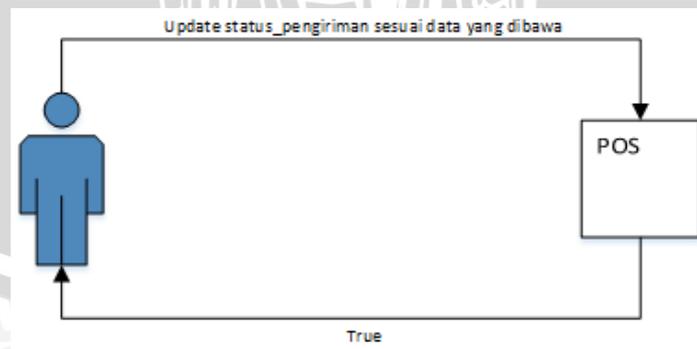




Gambar 4.11 Diagram alur mengambil data dari kantor

Pada Gambar 4.11 menunjukkan alur dari mekanisme pendaki mengambil data ACK dari server kantor yang dilakukan secara *background service*. Karena berjalan secara *background service* maka proses ini akan terjadi tanpa harus melibatkan pendaki.

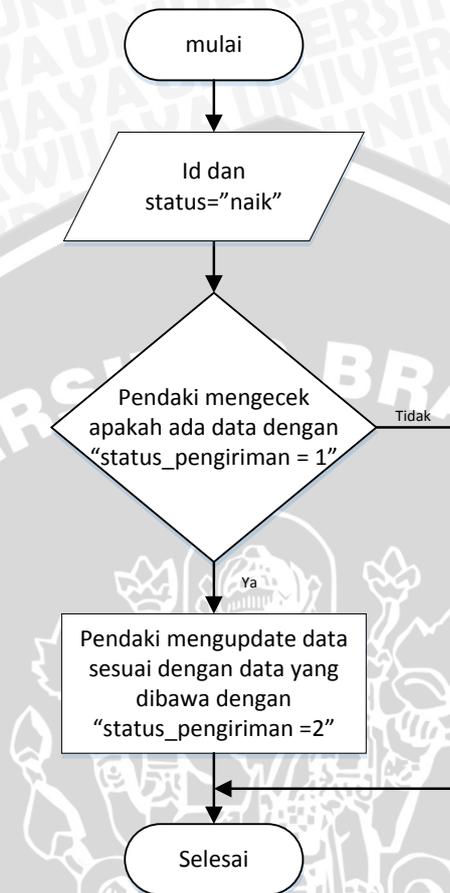
6. Pendaki mengirimkan ACK yang diterima ke setiap pos.



Gambar 4.12 Arsitektur pendaki mengupdate status

Pada tahap ini sesuai dengan Gambar 4.12, pendaki akan mengupdate data yang terdapat di setiap pos sesuai dengan data yang telah dibawa dari kantor. Data yang diupdate adalah data dengan status pengiriman bernilai 1 dirubah menjadi data dengan status pengiriman

bernilai 2. Disini merupakan cara untuk menyampaikan ACK dari kantor bahwa data yang dititipkan pada pendaki turun sudah diterima oleh kantor. Untuk alur skenario ini dapat dilihat pada Gambar 4.13.



Gambar 4.13 Diagram alur pendaki mengupdate status

Pada Gambar 4.13 menunjukkan alur dari mekanisme pendaki mengupdate nilai status pengiriman pada data server pos sesuai dengan data yang dibawa dari server kantor secara *background service*. Karena berjalan secara *background service* maka proses ini akan terjadi tanpa harus melibatkan pendaki.

4.3.2 Kebutuhan Non-fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang mempengaruhi jalannya sistem namun tidak ada secara langsung berhubungan dengan aplikasi. Dalam sistem ini kebutuhan non-fungsional yang dimiliki yakni :

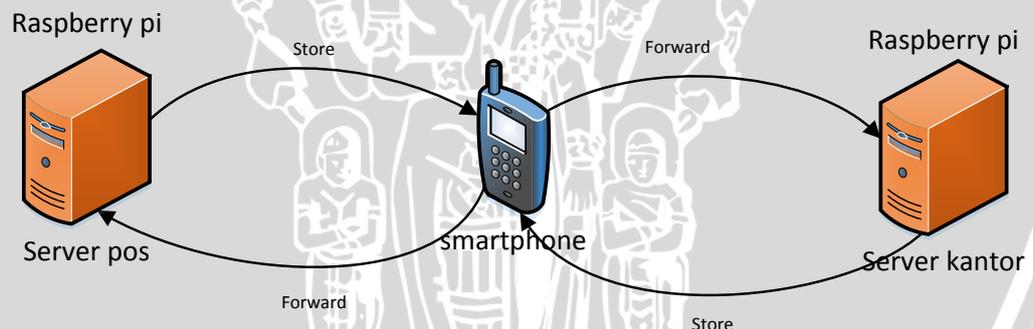
1. Sistem mampu melakukan pengiriman data dengan benar menggunakan metode *store and forward*.
2. Sistem ini memiliki *Raspberry pi* berperan sebagai *access point*.
3. Sistem ini memiliki *web service* yang menjadi penghubung antara *web server* dengan aplikasi android.



4. Dalam sistem ini pendaki cukup menyambungkan wifi smartphone miliknya dengan access point yang telah disediakan.
5. Aplikasi android ini merupakan aplikasi ringan yang akan mudah digunakan oleh tiap pendaki.
6. Pendaki tidak akan merasa terbebani karena hanya cukup menjalankan aplikasi di smartphone miliknya dan memasukkan ID satu kali di awal menjalankan.

4.4 Simulasi Metode *Store and forward*

Simulasi jaringan untuk pengiriman data menggunakan metode *store and forward* akan ditunjukkan pada Gambar 4.14. Pada gambar dapat dilihat bahwa ketika *web server* pos akan mengirimkan data kepada server kantor membutuhkan node perantara yang diperankan oleh smartphone milik pendaki. Begitu juga server kantor jika hendak mengirimkan data kepada *web server* pos maka membutuhkan node perantara untuk membantu mengirimkan. Pada pengiriman data ini hanya akan dilakukan komunikasi antara pendaki sebagai kurir dengan node sebagai server. Dalam sistem ini tidak dapat melakukan komunikasi secara langsung antara node server dengan server yang lain dikarenakan metode yang digunakan yakni metode *store and forward*. Dimana untuk berkomunikasi dengan node lainnya harus melalui node perantara atau kurir terlebih dahulu dimana dalam penelitian ini kurir diperankan oleh pendaki.



Gambar 4.14 Simulasi Metode *Store and forward*

Pada penelitian ini penulis menggambarkan terdapat tiga buah node seperti pada Gambar 4.14 untuk pengiriman data menggunakan metode *store and forward*, yaitu :

1. Server Pos

Server pos merupakan server yang diperankan oleh *Raspberry pi* dan kemudian telah diinstall operasi sistem raspbian serta telah dibentuk *web server*, *web service* serta *database* server di dalamnya sehingga siap menyimpan setiap ID pendaki yang melewatinya ke dalam *database* yang dimiliki. Adapun untuk melengkapi agar dapat berkomunikasi dengan pendaki yakni telah dibuat *access point* di dalam *Raspberry pi* dan

kemudian setiap *Raspberry pi* dinyalakan maka *access point* akan secara langsung berstatus ENABLED dimana itu berarti dia telah aktif. Ketika pendaki telah berada di area pos maka ketika smartphone miliknya tetap mengaktifkan wifi akan secara langsung terhubung dengan *access point* yang ada di dalam pos. Selanjutnya aplikasi yang ada di dalam smartphone akan secara langsung melakukan perilaku yang harus dilakukan. Server pos akan berperan sebagai *source* saat pos menyiapkan data yang akan dititipkan kepada pendaki untuk diberikan ke server kantor. Serta server pos juga dapat berperan sebagai *destination* ketika server kantor telah mengirimkan data melalui pendaki untuk diberikan kepada server pos.

2. Server Kantor

Server kantor merupakan server yang berada di kantor petugas dan diperankan oleh *Raspberry pi*. *Raspberry* ini telah diinstall operasi sistem raspbian dan kemudian dilengkapi dengan *web server*, *web service* serta *database* server di dalamnya. Dengan begitu server kantor telah siap untuk menerima data setiap pendaki ke dalam *database* yang telah dibangun. Untuk melengkapi jalan komunikasi antara pendaki dan server kantor maka diperlukan *access point* yang dibangun di dalamnya. Setelah itu maka pendaki akan secara langsung terhubung dengan server kantor setiap kali smartphone miliknya masuk ke area kantor dengan status wifi tetap aktif. Kemudian aplikasi pendaki akan secara *background service* melakukan perilaku yang harus dilakukan di area kantor sesuai dengan skenario. Server kantor akan berperan sebagai *source* ketika server kantor sedang mengirimkan data ACK yang akan diberikan kepada server pos melalui smartphone pendaki. Dan selanjutnya server kantor akan berperan sebagai *destination* ketika server pos melakukan pengiriman data melalui pendaki untuk ditujukan kepada server pos.

3. Smartphone

Dalam penelitian ini smartphone berperan sebagai node perantara atau kurir yang siap menyalurkan data yang sedang dikirimkan. Melalui smartphone maka data yang dikirimkan akan dapat sampai pada *destination* yang diinginkan. Smartphone disini merupakan smartphone android yang telah memiliki aplikasi pendakian khusus sebagai aplikasi utama dari penelitian ini serta wifi yang telah tersambung dengan *access point* yang disediakan yakni bernama "RASPI". Pendaki tidak akan merasa terbebani karena pendaki hanya cukup melakukan insert ID pendaki dan selanjutnya perilaku akan dilakukan oleh aplikasi secara *background service*.

Ketika pendaki telah berada di area server pos maka secara *background service* aplikasi pendakian yang dimiliki oleh pendaki akan melakukan perilaku yang seharusnya dilakukan tanpa sepengetahuan pendaki. Perilaku ini yakni memasukkan ID dengan status "naik" serta

mengupdate status pengiriman sesuai dengan data yang telah dititipkan dari server kantor bagi pendaki yang sedang melakukan pendakian. Perilaku lainnya yang dilakukan oleh aplikasi pendakian ketika pendaki berada pada area kantor maka yang dapat dilakukan aplikasi adalah mengambil data yang sudah disiapkan oleh server kantor dan memasukkan data yang telah dititipkan oleh server pos untuk diberikan kepada server kantor.

Pada simulasi pengiriman data menggunakan metode *store and forward* pada penelitian ini mempunyai dua arah pengiriman. Pengiriman pertama yakni pengiriman data ACK yang dilakukan oleh server kantor yang ditujukan kepada server pos. Dan yang kedua yakni pengiriman data pendaki dari setiap pos yang ditujukan kepada server kantor. Adapun keduanya dilakukan dengan bantuan aplikasi pendakian yang dimiliki oleh pendaki. Adapun aplikasi ini yang bertugas sebagai node perantara atau kurir yang bertugas untuk mengantarkan data hingga sampai pada tujuan. Sesuai dengan konsep pengiriman yang diambil yaitu metode *store and forward* maka semua pengiriman akan dapat sampai pada tujuan dengan menggunakan node perantara sebagai kurir pengiriman data dan pada penelitian ini diperankan oleh smartphone pendaki.

Adapun data pendaki yang berada di dalam server kantor dan server pos memiliki status pengiriman sebagai penanda apakah data telah terkirim sesuai scenario. Adapun penjelasan dari status pengiriman adalah sebagai berikut :

- Database server kantor

Tabel 4.1 Keterangan Status Data pada Server Kantor

Status_pengiriman	Keterangan
0	Dimana data yang memiliki nilai status_pengiriman ini merupakan data yang belum pernah diambil oleh pendaki untuk ditujukan ke server pos.
1	Dimana data yang memiliki nilai status_pengiriman ini merupakan data yang telah diambil oleh pendaki untuk ditujukan ke server pos.

- Database server pos

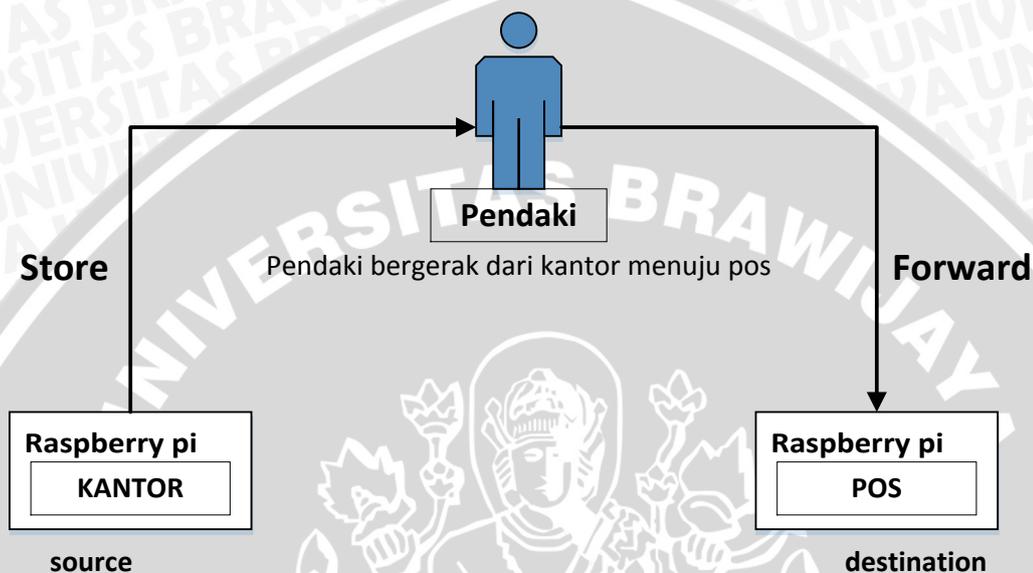
Tabel 4.2 Keterangan Status Data pada Server Pos

Status_pengiriman	Keterangan
0	Dimana data yang memiliki nilai status_pengiriman ini merupakan data yang belum pernah diambil oleh pendaki untuk ditujukan ke server kantor.
1	Dimana data yang memiliki nilai status_pengiriman ini merupakan data yang telah diambil oleh pendaki untuk ditujukan ke server kantor namun belum



	diketahui apakah data telah sampai atau belum.
2	Dimana data yang memiliki nilai status_pengiriman ini merupakan data yang telah diambil oleh pendaki dan telah sampai pada server kantor.

4.4.2 Mekanisme Pengiriman Data ACK dari Kantor ke Pos

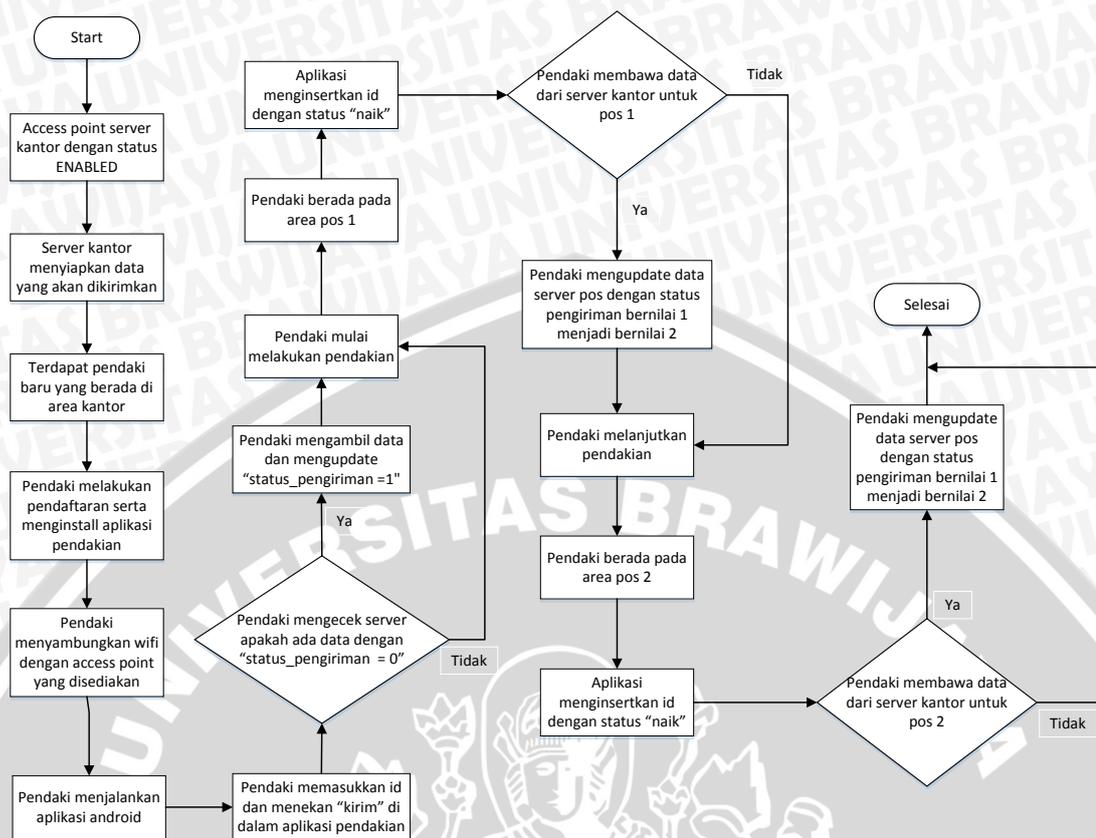


Gambar 4.15 Tampilan Pengiriman Data ACK ke Server Pos

Pada Gambar 4.15 menampilkan mekanisme pengiriman data ACK yang dilakukan oleh server kantor untuk ditujukan kepada setiap pos dengan tujuan memberitahu server pos bahwa data yang dikirimkan telah diterima oleh server kantor. Adapun data yang dikirimkan oleh server pos ini merupakan data yang sama dengan yang dikirimkan oleh server pos yang sudah sampai pada server kantor dan merupakan data yang belum pernah dikirimkan sebelumnya. Data yang belum pernah dikirimkan sebelumnya dapat dilihat dari status pengiriman yang dimiliki oleh data apakah masih bernilai 0, jika masih bernilai 0 maka itu berarti bahwa data tersebut belum pernah dikirimkan sebelumnya namun jika status pengiriman sudah bernilai 1 maka itu merupakan data yang sudah pernah dikirimkan sebelumnya. Data yang sudah bernilai 1 tidak akan dapat diambil oleh pendaki untuk dikirimkan kepada server pos.

Data yang dibawa oleh pendaki dari server pos merupakan data yang sama dimiliki oleh server pos sesuai dengan nama posnya dengan status pengiriman bernilai 1. Data yang bernilai 1 dari setiap pos ini akan diupdate oleh pendaki disesuaikan dengan data dari server kantor dengan status pengiriman menjadi bernilai 2. Data yang sudah bernilai 2 merupakan data pendaki yang dikirimkan setiap server pos untuk ditujukan ke server kantor dan sudah diterima oleh server kantor.

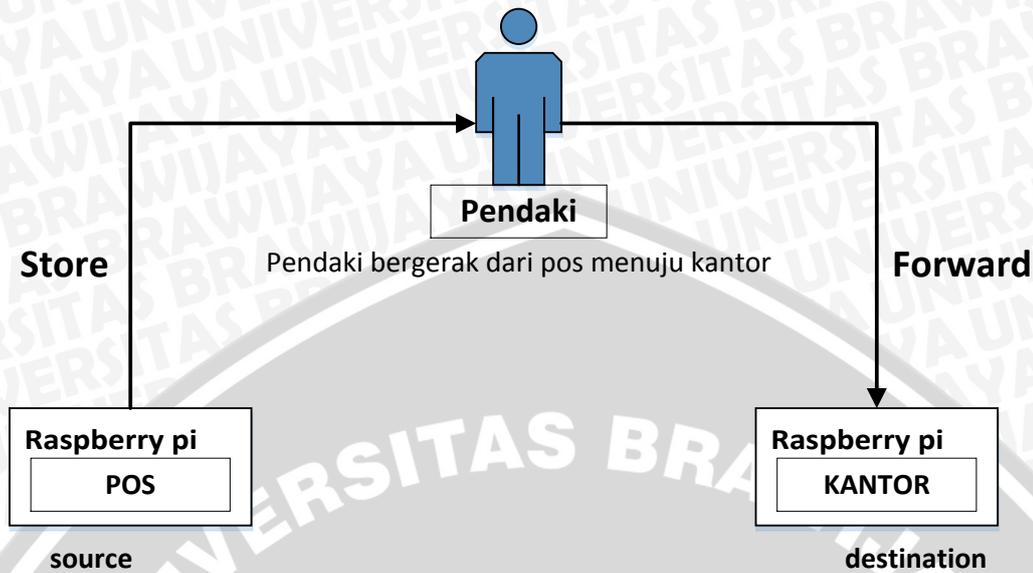




Gambar 4.16 Tampilan Alur Pengiriman ACK ke Server Pos

Pada Gambar 4.16 menampilkan alur pengiriman data ACK dari server kantor menuju server pos lebih lengkapnya. Pada skenario pengiriman disini penulis membuat server pos dengan jumlah dua pos dan satu server kantor. Adapun status pengiriman pada data di dalam server pos menunjukkan apakah kondisi dari data apakah data merupakan data yang baru masuk dengan ditandai status pengiriman bernilai 0, atau data yang sedang dikirim dengan maksud sedang dibawa oleh pendaki turun untuk dibawa ke server kantor ditandai dengan status pengiriman bernilai 1, dan yang terakhir ataukah data kondisi sudah terkirim dengan ditandai status pengiriman bernilai 2. Sedangkan status pengiriman di dalam server kantor menunjukkan apakah kondisi data merupakan data baru yang belum pernah dibawa oleh pendaki dengan ditandai status pengiriman bernilai 0, ataukah kondisi data telah diambil oleh pendaki yang ditandai dengan status pengiriman bernilai 1. Saat pendaki melakukan pengambilan data maka data yang telah diambil akan secara langsung mengupdate status pengiriman yang dimiliki menjadi bernilai 1 dengan tujuan sebagai tanda bahwa data tersebut sudah tidak bisa diambil lagi.

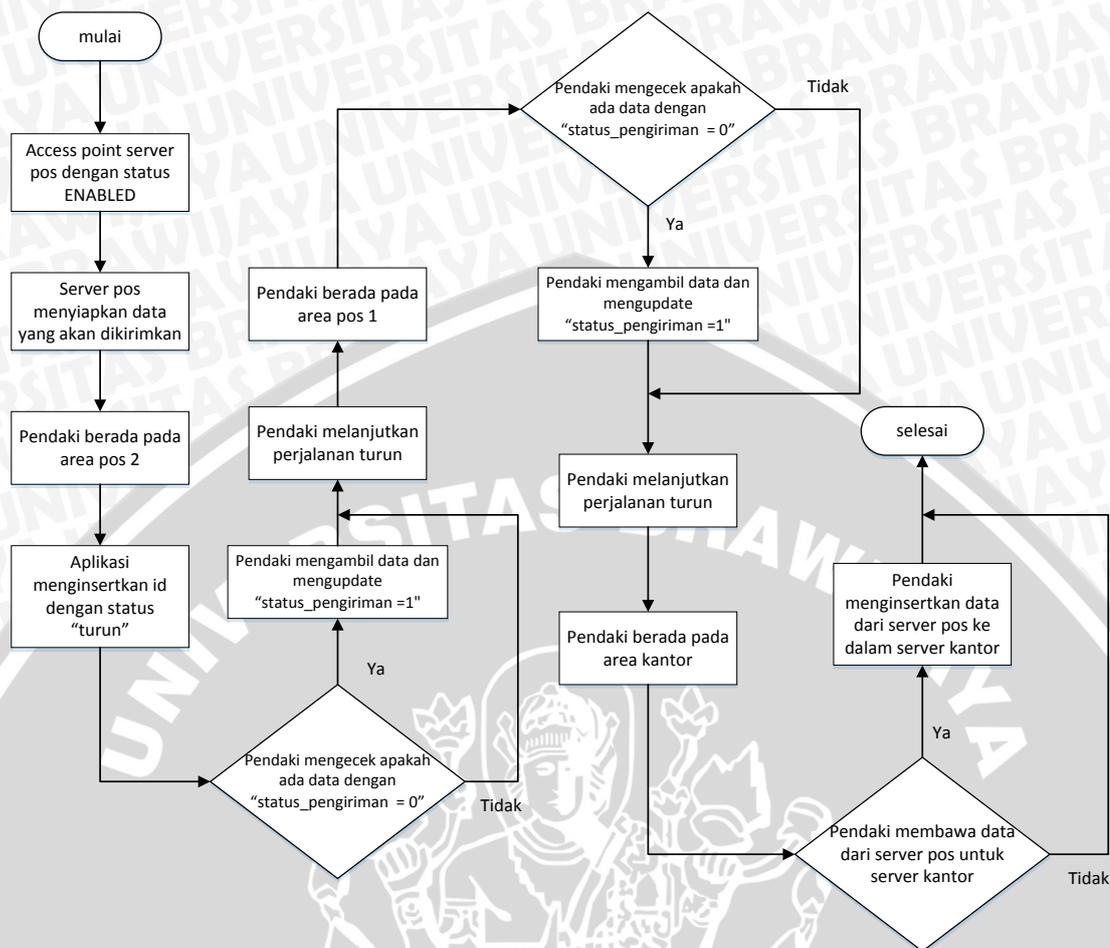
4.4.3 Mekanisme Pengiriman Data Titip dari Server Pos ke Server Kantor



Gambar 4.17 Tampilan Pengiriman Data Titip ke Server Kantor

Pada Gambar 4.17 menampilkan mekanisme pengiriman data pendaki yang ditiptkan oleh server pos untuk ditujukan pada server kantor. Adapun data disini merupakan data ID pendaki dilengkapi dengan waktu dan status saat pendaki berada dalam area pos. Pengiriman ini dilakukan oleh setiap pos untuk melaporkan kepada server kantor pendaki dengan ID apa saja yang telah melewati pos tersebut. Jika data berasal dari server pos 1 maka data yang dikirimkan merupakan data pendaki yang telah melewati pos 1 dengan dilengkapi waktu dan status pendaki apakah “naik” atau “turun”. Serta data yang disiapkan oleh server pos untuk dikirimkan ke server kantor adalah data yang memiliki status pengiriman bernilai 0. Dengan adanya data yang status pengiriman bernilai 0 dalam server pos maka aplikasi pendaki akan secara *background service* mengambil data tersebut dan kemudian mengupdate status pengiriman data yang telah diambil menjadi bernilai 1.

Dalam pengiriman data yang dilakukan oleh server pos dengan destination server kantor dapat dilakukan atas bantuan node perantara atau kurir yang diperankan oleh smartphone pendaki. Smartphone pendaki yang telah diinstall aplikasi pendakian dan tetap berada dalam jangkauan wifi lokal yang telah dibangun.



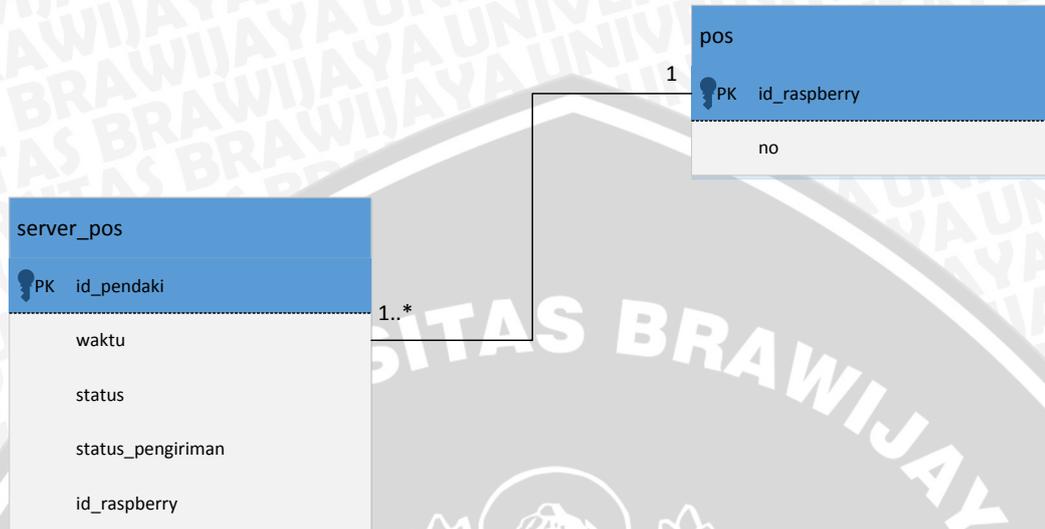
Gambar 4.18 Tampilan Alur Pengiriman Data Titip ke Server Kantor

Pada Gambar 4.18 menampilkan alur pengiriman data pendaki dari server pos untuk ditujukan kepada server kantor. Pengiriman data ini menggunakan metode *store and forward* yang berarti membutuhkan node perantara atau kurir yang dibutuhkan untuk menyalurkan dan disini diperankan oleh smartphone pendaki. Ketika pendaki berada pada area pos maka aplikasi akan secara *background service* mengambil data yang telah disiapkan oleh server pos untuk dibawa turun ke server kantor. Server pos akan menyiapkan data yang memiliki status pengiriman bernilai 0 untuk dikirimkan kepada server kantor. Setelah aplikasi pendakian mengambil data yang memiliki status pengiriman bernilai 0 maka selanjutnya status pengiriman dari data yang diambil akan berubah menjadi bernilai 1. Tujuannya yakni jika ada pendaki lain yang berada pada area pos maka tidak akan mengambil data yang sama yang pernah diambil oleh pendaki lain dengan begitu tidak akan terjadi penumpukan data.

4.5 Perancangan Database

Pada tahap ini penulis akan menjelaskan rancangan database yang telah dibuat yakni *database* yang akan digunakan pada server pos, server kantor, dan *database* di dalam aplikasi android. Setiap pos memiliki *database* untuk

menyimpan ID pendaki yang telah dikirimkan oleh setiap pendaki. Pada *database* tersebut juga dilengkapi dengan no, ID raspberry, waktu, status, serta status pengiriman. *Database* ini bernama serverpos. Adapun rancangan tersebut dapat dilihat pada Gambar 4.19



Gambar 4.19 Rancangan *Database* Pada Server Pos

Pada database serverpos memiliki dua tabel yakni tabel pos dan tabel server_pos. adapun penjelasannya adalah sebagai berikut :

- Tabel pos yakni berisikan *field* id_raspberry dan no. Adapun penjelasannya dapat dilihat pada Tabel 4.3.

Tabel 4.3 Keterangan Kolom Tabel pos pada serverpos

Nama <i>Field</i>	Deskripsi
no	Berisi nomor urut data yang masuk ke database
Id_raspberry	Berisi ID pos sesuai dengan data identitas pos

- Tabel server_pos memiliki beberapa *field* yakni id_pendaki, waktu, status, status_pengiriman, dan id_raspberry. Adapun penjelasannya dapat dilihat pada Tabel 4.4.

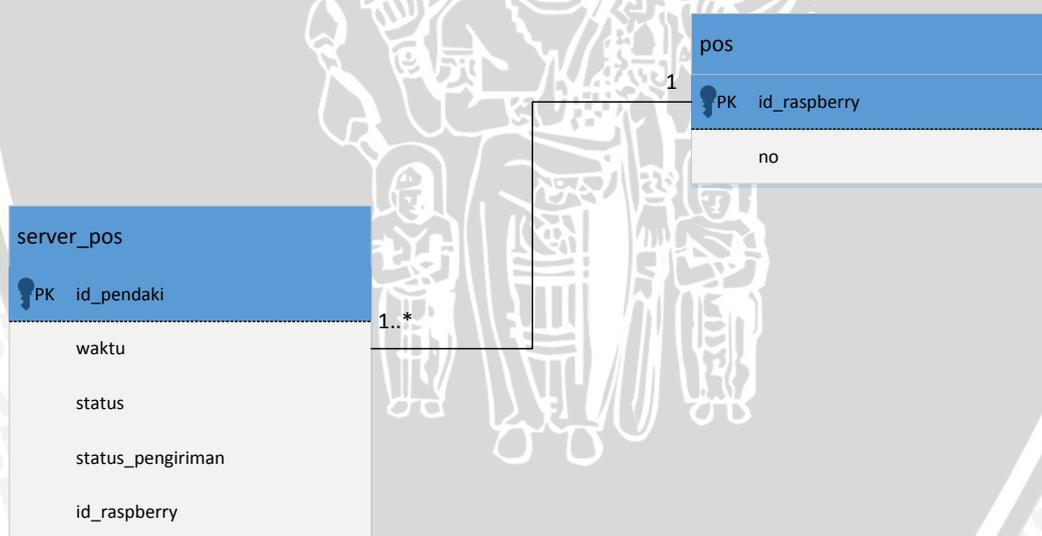
Tabel 4.4 Keterangan Kolom Tabel server_pos pada serverpos

Nama <i>Field</i>	Deskripsi
Id_raspberry	Berisi ID pos sesuai dengan identitas pos
Id_pendaki	Berisi ID pendaki
waktu	Berisi waktu sesuai dengan data masuk
status	Berisi status sesuai dengan status pendaki
status_pengiriman	Berisi status pengiriman sesuai dengan status data

Adapun pada *field* status pengiriman dimana isi dari status pengiriman ini ada tiga macam nilai yakni :

- Status pengiriman bernilai 0 dimana data ini merupakan data baru dan belum pernah diambil oleh pendaki atau dikirimkan kepada server kantor. Data dengan status pengiriman bernilai 0 ini akan disiapkan oleh server pos untuk dikirimkan ke server kantor.
- Status pengiriman bernilai 1 dimana data ini merupakan data yang telah diambil oleh pendaki namun belum sampai pada server kantor. Data ini sudah tidak dapat diambil lagi atau dikirimkan kembali.
- Status pengiriman bernilai 2 dimana data ini merupakan status data yang telah diterima oleh server kantor.

Adapun server kantor memiliki *database* untuk menyimpan ID pendaki yang telah dikirimkan oleh setiap pendaki. Pada *database* tersebut juga dilengkapi dengan no, ID raspberry, waktu, status, serta status pengiriman. *Database* ini bernama serverakhir. Adapun rancangan tersebut dapat dilihat pada Gambar 4.20



Gambar 4.20 Rancangan Database pada Server Kantor

Pada database serverakhir memiliki dua tabel yakni tabel pos dan tabel server_pos. adapun penjelasannya adalah sebagai berikut :

- Tabel pos yakni berisikan *field* id_raspberry dan no. Adapun penjelasannya dapat dilihat pada Tabel 4.5.



Tabel 4.5 Keterangan Kolom Tabel pos pada serverakhir

Nama <i>Field</i>	Deskripsi
no	Berisi nomor urut data yang masuk ke database
Id_raspberry	Berisi ID pos sesuai dengan data yang dikirimkan pendaki

- Tabel server_pos memiliki beberapa *field* yakni id_pendaki, waktu, status, status_pengiriman, dan id_raspberry. Adapun penjelasannya dapat dilihat pada Tabel 4.6.

Tabel 4.6 Keterangan Kolom Tabel server_pos pada serverakhir

Nama <i>Field</i>	Deskripsi
Id_raspberry	Berisi ID pos sesuai dengan data yang dikirimkan pendaki
Id_pendaki	Berisi ID pendaki sesuai dengan data yang dikirimkan pendaki
waktu	Berisi waktu sesuai dengan data yang dikirimkan pendaki
status	Berisi status sesuai dengan data yang dikirimkan pendaki
status_pengiriman	Berisi status pengiriman sesuai dengan status data

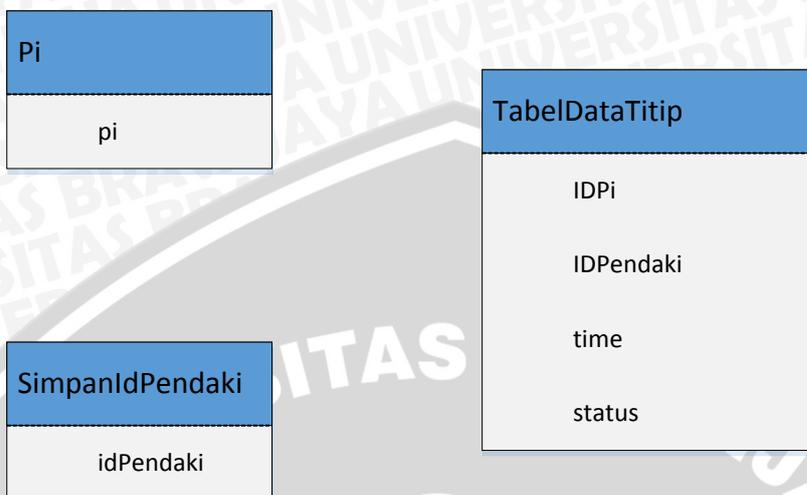
Adapun pada *field* status pengiriman dimana isi dari status pengiriman ini ada dua macam nilai yakni :

- o Status pengiriman bernilai 0 dimana data ini merupakan data baru dan belum pernah dititipkan kepada pendaki untuk dibawa ke server pos. Data ini merupakan data yang disiapkan oleh server kantor untuk diambil oleh pendaki dan dibawa ke server pos.
- o Status pengiriman bernilai 1 dimana data ini merupakan data yang sudah dibawa oleh pendaki dan ditujukan ke server pos. Data yang sudah memiliki status pengiriman bernilai 1 maka tidak dapat lagi diambil oleh pendaki.

Untuk *database* pada aplikasi android memiliki rancangan seperti pada Gambar 4.21. Dimana aplikasi android dirancang memiliki tiga tabel yaitu Pi, SimpanIdPendaki, dan TabelDataTitip. Adapun penjelasan fungsi dari masing – masing *database* adalah :

- Tabel Pi dirancang untuk menyimpan id_raspberry atau identitas pos dimana pendaki berada.

- Tabel SimpanIdPendaki dirancang untuk menyimpan ID pendaki sendiri.
- Tabel TabelDataTitip dirancang untuk menyimpan data yang dititipkan kepada pendaki.



Gambar 4.21 Rancangan *database* pada aplikasi android

Gambar 4.21 menunjukkan rancangan *database* di dalam aplikasi android yang akan digunakan oleh pendaki. Di dalam aplikasi ini menggunakan *realm database*.

4.6 Struktur Pengiriman File JSON

Dalam pengiriman data pada sistem ini menggunakan format JSON karena akan lebih mudah dibaca dan sangat praktis. Terdapat dua tahap dari skenario yang menggunakan format JSON di dalamnya. Tahap dimana pendaki akan mengambil data yang siap dititipkan dari setiap pos untuk dibawa turun ke server kantor dan tahap dimana pendaki mengambil data ACK yang telah disiapkan oleh server kantor untuk dibawa ke setiap pos dengan tujuan mengupdate status pengirimannya. Struktur konten penamaan pada tahap pendaki mengambil data dari setiap pos dapat dilihat pada Tabel 4.7. Sedangkan struktur konten penamaan pada tahap pendaki mengambil data dari server kantor dapat dilihat pada Tabel 4.8.

Tabel 4.7 Struktur Konten Ambil Data Dari Pos

Elemen Penamaan	Deskripsi
No	Berisi nomor sesuai dengan data yang dikirim
Id_raspberry	Berisi ID pos sesuai dengan data yang dikirim
Id_pendaki	Berisi ID pendaki sesuai dengan data yang dikirim
waktu	Berisi waktu sesuai dengan data yang dikirim
status	Berisi status sesuai dengan data yang dikirim

status_pengiriman	Berisi status pengiriman sesuai dengan data yang dikirim
-------------------	--

Pada Tabel 4.7 menunjukkan struktur konten JSON dimana telah dijelaskan isi dari setiap elemennya. Setiap pengambilan data dari server pos maka akan membentuk satu konten JSON yang terdiri dari beberapa elemen diatas.

Tabel 4.8 Struktur Konten Ambil Data Dari Kantor

Elemen Penamaan	Deskripsi
No	Berisi nomor sesuai dengan data yang dikirim
id_raspberry	Berisi ID pos sesuai dengan data yang dikirim
id_pendaki	Berisi ID pendaki sesuai dengan data yang dikirim
waktu	Berisi waktu sesuai dengan data yang dikirim
Status	Berisi status sesuai dengan data yang dikirim

Pada Tabel 4.8 menunjukkan struktur konten JSON dimana telah dijelaskan isi dari setiap elemennya. Setiap pengambilan data dari server kantor maka akan membentuk satu konten JSON yang terdiri dari beberapa elemen diatas.



4.7 Perancangan Antarmuka Aplikasi Android

Perancangan antarmuka pada aplikasi android dapat dilihat pada Gambar 4.22. Pada gambar terlihat rancangan antarmuka pada aplikasi android yang hanya memiliki satu field untuk menginputkan ID pendaki dan tombol “kirim” untuk mengirimkan id.



Gambar 4.22 Rancangan antarmuka aplikasi android

Pada Gambar 4.22 menunjukkan tampilan dari aplikasi android yang akan dibuat. Pada aplikasi ini penulis tidak memfokuskan pada desain antarmuka melainkan *background service* yang akan membantu pendaki menggunakan aplikasi dengan mudah.

BAB 5 IMPLEMENTASI

Pada bab ini akan menjelaskan implementasi sistem yang dibangun dengan berdasarkan pada perancangan sistem sebelumnya. Proses implementasi yakni dilakukan dalam merelisasikan rancangan sistem serta mengetahui kesesuaian sistem untuk menyelesaikan masalah yang telah ditentukan.

5.1 Implementasi Simulasi Jaringan

Pada bab perancangan telah dijelaskan bagaimana akan dibangun sebuah sistem untuk mengimplementasikan metode *store and forward* pada sistem pemantauan pendaki gunung. Langkah yang dilakukan penulis adalah menyiapkan *Raspberry pi* untuk berperan sebagai pos serta berperan sebagai server kantor, serta smartphone untuk berperan sebagai perantara dalam pengiriman data.

Langkah selanjutnya adalah pada raspberry dan linux akan diinstall Apache *web server*, MySQL serta *web service*. *Database* untuk setiap server akan menggunakan MySQL seperti yang dijelaskan pada bab perancangan. Setelah instalasi tersebut telah selesai maka selanjutnya akan melakukan konfigurasi agar dapat membangun *access point* di dalam *Raspberry pi*. *Access point* ini akan menjadi jaringan lokal yang menghubungkan *Raspberry pi* dengan smartphone agar dapat mengirimkan data.

5.2 Konfigurasi *web server* dan alamat ip

Dalam tahap ini akan membahas konfigurasi untuk *web server* serta alamat ip yang digunakan oleh penulis. Langkah pertama yang dilakukan adalah konfigurasi untuk mengatur ip static untuk eth0. Adapun isi konfigurasi dapat dilihat pada Tabel 5.1.

Tabel 5.1 Konfigurasi ip static eth0

No	etc/network/interfaces
1	auto eth0
2	iface eth0 inet static
3	address 192.168.137.10
4	netmask 255.255.255.0
5	network 192.168.137.0
6	broadcast 192.168.137.255
7	gateway 192.168.137.1

Setelah konfigurasi eth0 sudah berhasil maka langkah selanjutnya adalah menginstall Apache2 untuk sebagai *web server*nya.

```
Sudo apt-get install apache2 -y
```

Langkah selanjutnya adalah instalasi php dan paketnya apache untuk melengkapi pembuatan *web server*.

```
sudo apt-get install php5 libapache2-mod-php5 -y
```

Kemudian melakukan langkah terakhir yaitu konfigurasi untuk *database* dalam server *Raspberry pi*. Dalam server ini menggunakan MySQL sebagai *databasenya*, dalam instalasi dapat dilihat sebagai berikut :

```
sudo apt-get install mysql-server php5-mysql -y
```

Setelah berhasil diinstall maka kita mulai dengan restart Apache dengan perintah :

```
sudo service apache2 restart
```

Saat semuanya telah siap maka penulis melanjutkan ke tahap berikutnya yaitu menjadikan *raspberry* sebagai *wifi access point* sehingga dapat memberikan jaringan internet lokal untuk *smartphone* pendaki. Adapun konfigurasi dalam tahap ini akan dibahas pada sub bab berikutnya.

5.3 Konfigurasi menjadikan *Raspberry pi* sebagai *wifi access point*

Pada sub bab ini akan menjelaskan konfigurasi untuk menjadikan *Raspberry pi* sebagai *wifi access point*. Tahap awal yang dilakukan adalah menginstall *hostapd* dan *isc-dhcp-server*. *Hostapd* digunakan sebagai aplikasi *wifi router* untuk linux sedangkan untuk setting DHCP server pada linux menggunakan *isc-dhcp-server*. Setelah instalasi berhasil maka selanjutnya membangun DHCP Server, adapun langkahnya yaitu :

- Langkah pertama, menuliskan perintah untuk setting DHCP server

```
sudo nano /etc/dhcp/dhcp.conf
```

Selanjutnya di dalam file tersebut menonaktifkan baris berikut :

```
option domain-name "example.org";  
option domain-name-servers ns1.example.org,  
ns2.example.org;
```

Dan mengaktifkan baris berikut :

```
authoritative;
```

Kemudian menuliskan konfigurasi untuk routernya dapat dilihat pada Tabel 5.2.

Tabel 5.2 Konfigurasi untuk range router

No	/etc/dhcp/dhcp.conf
1	subnet 192.168.42.0 netmask 255.255.255.0 {
2	range 192.168.42.10 192.168.42.50;
3	option broadcast-address 192.168.42.255;
4	option routers 192.168.42.1;
5	default-lease-time 600;

6	max-lease-time 7200;
7	option domain-name "local";
8	option domain-name-servers 8.8.8.8, 8.8.4.4;
9	}

Kemudian kita mengatur interface yang digunakan untuk dhcp server, yaitu wlan0. Adapun konfigurasinya dapat dilihat pada Gambar 5.1.

```

GNU nano 2.2.6      File: /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server initscript
# sourced by /etc/init.d/isc-dhcp-server
# installed at /etc/default/isc-dhcp-server by the maintainer scripts
#
# This is a POSIX shell fragment
#
# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPD_CONF=/etc/dhcp/dhcpd.conf
#
# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPD_PID=/var/run/dhcpd.pid
#
# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""
#
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan0"
    
```

Gambar 5.1 Konfigurasi isc-dhcp-server

- Langkah kedua, melakukan mengatur agar wlan0 menjadi ip static. Adapun langkahnya sama seperti membuat ip static untuk eth0 dan dapat dilihat pada Tabel 5.3.

Tabel 5.3 Konfigurasi ip static wlan0

No	/etc/network/interfaces
1	allow-hotplug wlan0
2	iface wlan0 inet static
3	address 192.168.42.1
4	netmask 255.255.255.0

- Langkah terakhir adalah melakukan konfigurasi untuk *access point*nya. Dimulai dengan membuat file baru pada hostapd untuk nantinya dijalankan. Isi dari konfigurasi dapat dilihat pada Tabel 5.4.

Tabel 5.4 Konfigurasi access point

No	sudo nano /etc/hostapd/hostapd.conf
1	interface=wlan0
2	driver=rtl871xdrv
3	SSID=RASPI
4	hw_mode=g
5	channel=6



6	macaddr_acl=0
7	auth_algs=1
8	ignore_broadcast_SSID=0
9	rsn_pairwise=CCMP

Setelah konfigurasi telah selesai dibuat maka langkah selanjutnya adalah memanggil file konfigurasi tersebut di dalam file `hostapd` dengan mencari baris "`DAEMON_CONF=""`" dapat dilihat pada Tabel 5.5.

No	/etc/default/hostapd
1	DAEMON_CONF="/etc/hostapd/hostapd.conf"

- Untuk menjalankannya kita dapat menuliskan perintah seperti pada Tabel 5.5.

Tabel 5.5 Perintah menjalankan *access point*

```
sudo ifconfig wlan0 192.168.42.1/24 up
sleep 1
sudo service isc-dhcp-server restart
sudo service hostapd restart
sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
```

5.4 Implementasi *Store and forward*

Implementasi *store and forward* yakni proses dimana pengiriman data dapat dilakukan. Sistem ini menggunakan protokol HTTP yang dibangun diatas TCP dimana ini dapat menjamin data yang dikirimkan tidak akan rusak. Para pendaki akan dapat berkomunikasi dengan server menggunakan protokol HTTP serta pendaki akan secara *background service* menghubungi *Raspberry pi* dimana *web server* berada dan kemudian mengirimkan data yang akan dikirimkan.

Pada tahap ini yang berperan yakni pendaki, server pos serta server kantor. Adapun penjelasannya adalah :

- Pendaki berperan sebagai perantara atau node perantara dalam proses pengiriman data. Pengiriman data dilakukan oleh server pos untuk ditujukan pada server kantor dan juga server kantor untuk ditujukan pada server pos akan dilakukan melalui node perantara terlebih dahulu.
- Server pos yakni *web server* yang berkomunikasi dengan para pendaki. Dimana server pos dapat menerima ID pendaki yang telah dimasukkan ke dalam *database* miliknya dan kemudian server pos juga dapat menitipkan data kepada pendaki untuk ditujukan pada server kantor. Server pos

hanya dapat berkomunikasi dengan pendaki dan tidak dapat melakukan komunikasi secara langsung dengan server kantor.

- Server kantor yakni *web server* yang berkomunikasi dengan para pendaki dimana dapat menitipkan data kepada pendaki untuk dikirimkan kepada server pos. Serta server kantor dapat menerima data dari server pos yang dititipkan kepada pendaki ke dalam *database* miliknya. Server pos hanya dapat berkomunikasi dengan pendaki dan tidak bisa melakukan komunikasi secara langsung dengan server pos.

5.4.1 Implementasi Pendaki memasukkan ID dengan status “naik”

Pada tahap implementasi pendaki memasukkan ID dengan status “naik” merupakan skenario dimana pendaki yang sedang melakukan pendakian ke arah puncak akan mengirimkan ID pada setiap pos secara *background service* melalui aplikasi android. Ketika pendaki belum sampai pos puncak maka pendaki akan secara langsung mengirimkan status “naik”. Tabel 5.6 akan menampilkan potongan program untuk menjalankan skenario ini.

Tabel 5.6 Program InsertNaik.php

No	File : insertNaik.php
1	<?php
2	\$pos = 1; // identitas pos
3	require_once("koneksi.php");
4	
5	//Getting values
6	\$id_pendaki = \$_POST['id_pendaki'];
7	
8	//Creating an sql query
9	\$cekuser = mysql_query("SELECT * FROM server_pos
10	WHERE id_pendaki = '". \$id_pendaki.'" AND id_raspberry =
11	'\$pos' and status='naik');
12	if(mysql_num_rows(\$cekuser) < 1)
13	{
14	\$simpan = mysql_query("INSERT INTO
15	server_pos(id_raspberry,id_pendaki,status)
16	VALUES('". \$pos."', '\$id_pendaki', 'naik')");
17	if(\$simpan) {
18	echo \$pos;
19	} else {
20	echo "false";
21	}
22	}else{
23	echo \$pos;

24	}
25	?>

Program pada Tabel 5.6 telah dibuat sesuai dengan skenario yang telah dibuat yakni memasukkan ID dengan status “naik” selama pendaki belum mencapai pos puncak. Adapun penjelasan sourcecode adalah sebagai berikut :

- Baris 2 menunjukkan program akan lebih dulu mengenalkan pos berapa sekarang pendaki berada.
- Baris 3 untuk memanggil file koneksi.php.
- Baris 6 adalah inisialisasi \$id_pendaki.
- Baris 9 – 11 adalah untuk mengecek apakah di dalam *database* server sudah terdapat ID yang sama dengan milik pendaki.
- Baris 12 – 16 adalah jika di dalam *database* belum terdapat ID yang sama telah masuk maka proses menginsert ID akan dijalankan.
- Baris 17 – 21 menunjukkan jika menjalankan \$simpan maka akan menampilkan \$pos, namun jika tidak maka menampilkan “false”.
- Baris 22 – 24 menunjukkan jika ID sudah terdapat di dalam *database* maka hanya akan menampilkan \$pos.

5.4.2 Implementasi Pendaki memasukkan ID dengan status “turun”

Pada tahap implementasi pendaki memasukkan ID dengan status “turun” merupakan skenario dimana pendaki yang sedang melakukan pendakian ke arah bawah dan sudah melewati puncak akan mengirimkan ID pada setiap pos secara *background service* melalui aplikasi android miliknya. Ketika pendaki belum mencapai puncak dan sudah kembali turun maka proses ini tidak dapat dijalankan, namun saat sudah melewati pos puncak program ini akan berjalan secara langsung mengirimkan ID pendaki dengan status “turun”. Tabel 5.7 akan menampilkan potongan program untuk menjalankan skenario ini.

Tabel 5.7 program insertTurun.php

No	File : insertTurun.php
1	<?php
2	\$pos = 3;
3	
4	require_once("koneksi.php");
5	//Getting values
6	\$id_pendaki = \$_POST['id_pendaki'];
7	
8	//Creating an sql query
9	\$cekuser = mysql_query("SELECT * FROM server_pos
10	WHERE id_pendaki = '\$id_pendaki.'"
11	AND id_raspberry = '\$pos' and status='turun');"



12	<code>if(mysql_num_rows(\$cekuser) < 1)</code>
13	<code>{</code>
14	<code> \$simpan = mysql_query("INSERT INTO</code>
15	<code>server_pos(id_raspberry,id_pendaki,status)</code>
16	<code> VALUES('".\$\$pos."','".\$\$id_pendaki','turun'))");</code>
17	<code> if(\$simpan) {</code>
18	<code> echo \$pos;</code>
19	<code> } else {</code>
20	<code> echo "false";</code>
21	<code> }</code>
22	<code> }else{</code>
23	<code> echo \$pos;</code>
24	<code> }</code>
25	<code>?></code>

Program pada Tabel 5.7 telah dibuat sesuai dengan skenario yang telah dibuat yakni memasukkan ID dengan status “turun” setelah pendaki mencapai pos puncak. Adapun penjelasan sourcecode adalah sebagai berikut :

- Baris 2 yaitu program akan lebih dulu mengenalkan pos berapa sekarang pendaki berada.
- Baris 4 yaitu memanggil file koneksi.php.
- Baris 6 adalah inisialisasi \$id_pendaki.
- Baris 9 – 11 adalah pengecekan isi *database* server pos apakah sudah terdapat ID yang dimiliki oleh pendaki.
- Baris 12 – 16 yaitu jika di dalam *database* belum terdapat ID yang sama telah masuk maka proses menginsert ID akan dijalankan.
- Baris 17 – 21 jika menjalankan \$simpan maka menampilkan \$pos, namun jika tidak menjalankan \$simpan maka menampilkan “false”.
- Baris 22 – 24 menunjukkan jika ID sudah terdapat di dalam *database* maka hanya akan menampilkan \$pos.

5.4.3 Implementasi Pendaki turun mengambil data titip dari server pos

Pada implementasi pendaki turun mengambil data titip dari server pos merupakan skenario dimana pendaki yang turun setelah melewati pos puncak akan mengambil data yang sudah disiapkan oleh pos untuk dititipkan dan dibawa ke server kantor. Tahap ini dilakukan oleh *background service* android dengan tanpa sepengetahuan pendaki. Awalnya saat pendaki sedang memasukkan ID dengan status “turun” maka ketika itu juga aplikasi akan mengecek apakah terdapat data yang siap untuk diambil olehnya. Data yang siap dititipkan yakni yang data yang mempunyai status pengiriman bernilai 0. Data tersebut berarti data baru yang belum pernah dititipkan kepada pendaki yang lain. Data yang

sudah diambil oleh pendaki akan ditandai dengan status pendaki berubah bernilai 1. Jika data mempunyai status pengiriman bernilai 1 maka aplikasi pendaki tidak akan mengambilnya karena dianggap bukan sebagai data yang siap dititipkan. Status pengiriman bernilai 1 berarti bahwa data tersebut telah dititipkan kepada pendaki untuk dibawa ke server kantor. Adapun potongan program yang menjalankan ini dapat dilihat pada Tabel 5.8.

Tabel 5.8 program ngambilDataTurun.php

No	File : ngambilDataTurun.php
1	<?php
2	include "koneksi.php";
3	\$query = mysql_query ("select * from server_pos where
4	status_pengiriman='0'");
5	
6	\$respon=array();
7	while (\$row=mysql_fetch_array(\$query)){
8	\$item=array();
9	\$item["No"]=\$row["No"];
10	\$item["id_raspberry"]=\$row["id_raspberry"];
11	\$item["id_pendaki"]=\$row["id_pendaki"];
12	\$item["waktu"]=\$row["waktu"];
13	\$item["status"]=\$row["status"];
14	\$item["status_pengiriman"]=\$row["status_pengiriman"];
15	
16	mysql_query("update server_pos set status_pengiriman =
17	'1' where id_pendaki='".\$row['id_pendaki']."'");
18	array_push(\$respon,\$item);
19	}
20	\$response=array();
21	\$response["status"]=true;
22	\$response["data"]=\$respon;
23	echo json_encode(\$response);
24	?>

Pada Tabel 5.8 telah dibuat sesuai dengan skenario yakni dilakukan oleh pendaki yang sedang turun dan dilakukan secara *background service* ketika pendaki telah tiba di area pos. Adapun penjelasan kode program adalah sebagai berikut :

- Baris 2 yaitu memanggil file koneksi.php.
- Baris 3 – 4 yaitu program dengan mengecek apakah terdapat data dengan status pengiriman bernilai 0.
- Baris 6 adalah inisialisasi \$respon.

- Baris 7 – 19 dilanjutkan dengan perlakuan jika terdapat data dengan status pengiriman bernilai 0, maka akan melakukan perulangan di dalam array sebanyak data yang status pengiriman bernilai 0 telah diambil semua. Dan kemudian data yang sudah diambil akan diupdate dengan status pengiriman menjadi bernilai 1.
- Baris 20 adalah inisialisasi \$response.
- Baris 21 – 22 yaitu mengolah data dan kemudian dimasukkan ke dalam array bersarang.
- Baris 24 merupakan normalisasi data sebagai JSON.

5.4.4 Implementasi Pendaki turun memasukkan data titip ke server kantor

Pada implementasi pendaki turun memasukkan data titip ke server kantor merupakan skenario dimana pendaki turun yang telah membawa data dari setiap pos saat sedang perjalanan akan memasukkan data ke server kantor. Tahap ini merupakan tahap akhir bagi pendaki untuk menyampaikan pesan dari setiap pos kepada server kantor. Data yang dimasukkan ke server kantor dilakukan dengan *background service* oleh aplikasi android miliknya tanpa sepengetahuan pendaki. Ini dilakukan untuk memastikan data keberadaan setiap pendaki dapat diketahui oleh petugas yang berada di kantor dan dapat ditampilkan oleh petugas melalui monitor. Saat data yang dibawa telah masuk ke *database* server secara keseluruhan maka pengiriman data disebut berhasil dilakukan. Adapun potongan program dapat dilihat pada Tabel 5.9.

Tabel 5.9 Program insertserver.php

No	File : insertserver.php
1	<?php
2	include "koneksibawah.php";
3	\$id_pendaki = \$_POST['id_pendaki'];
4	\$id_raspberry = \$_POST['id_raspberry'];
5	\$waktu = \$_POST['waktu'];
6	\$status = \$_POST['status'];
7	\$status_pengiriman = \$_POST['status_pengiriman'];
8	\$checkData = mysql_query("select * from server_pos
9	where id_raspberry = '\$id_raspberry' and
10	id_pendaki = '\$id_pendaki' and
11	status = '\$status' and
12	status_pengiriman = '\$status_pengiriman' ");
13	if(mysql_num_rows(\$checkData) < 1){
14	
15	\$query=mysql_query("insert into server_pos
16	(id_raspberry,id_pendaki,waktu,status,status_pengiriman)



17	VALUE('\$sid_raspberry','\$sid_pendaki','\$waktu','\$status','\$status
18	_pengiriman')) or die (mysql_error());
19	
20	if (\$query){
21	echo "true";
22	} else {
23	echo "false";
24	}
25	}else{
26	echo false;
27	}
28	?>

Pada Tabel 5.9 merupakan program yang dijalankan ketika pendaki yang berstatus turun sudah sampai pada area kantor. Pendaki inilah yang melakukan secara *background service* memasukkan data yang telah dititipkan kepadanya ke dalam *database* server kantor. Adapun penjelasan kode program adalah sebagai berikut :

- Baris 2 memanggil file koneksi.php.
- Baris 3 mendeklarasikan variabel \$sid_pendaki.
- Baris 4 mendeklarasikan variabel \$sid_raspberry.
- Baris 5 mendeklarasikan variabel \$waktu.
- Baris 6 mendeklarasikan variabel \$status.
- Baris 7 mendeklarasikan variabel \$status_pengiriman.
- Baris 8 – 12 mengecek data apakah terdapat data yang sama dengan data yang sedang dibawa di dalam server kantor
- Baris 15 – 18 jika di dalam server kantor tidak ada data yang sama dengan data yang dibawa maka program akan melanjutkan program untuk memasukkan data ke *database* server kantor, namun jika terdapat data yang sama maka data tersebut tidak akan dimasukkan untuk kedua kalinya. Dengan begini tidak akan terdapat dua data yang sama di dalam *database* server kantor. Implementasi pendaki mengambil data titip dari server kantor

Pada implementasi pendaki mengambil data titip dari server kantor merupakan skenario dimana terdapat pendaki baru yang memulai pendakian dengan melakukan pendaftaran di kantor. Saat melakukan pendaftaran di kantor, petugas akan menjelaskan bahwa pendaki harus terhubung dengan wifi lokal yang telah disediakan dan menginstall aplikasi android di dalam smartphone miliknya serta memasukkan ID pendaki yang disediakan untuk dirinya. Saat pendaki telah terhubung dengan wifi yang lokal yang telah disediakan dan kemudian menjalankan aplikasinya maka secara langsung dengan

background service tanpa sepengetahuan pendaki akan mengambil data yang telah siap untuk dititipkan. Pendaki akan mengambil data ACK yang siap untuk dikirimkan ke setiap pos yang akan dilalui. Data ini merupakan isi dari *database* server dimana data memiliki status pengiriman bernilai 0. Data yang memiliki status pengiriman bernilai 0 berarti bahwa data itu telah disiapkan oleh server untuk dititipkan kepada pendaki dan kemudian dikirimkan kepada setiap pos yang akan dia lewati. Potongan program dapat dilihat pada Tabel 5.10.

Tabel 5.10 Program ngambilServerBawah.php

No	File : ngambilServerBawah.php
1	<?php
2	include "koneksibawah.php";
3	
4	\$query=mysql_query("select * from server_pos
5	where status_pengiriman='0'");
6	
7	if(mysql_num_rows(\$query) > 0){
8	\$respon=array();
9	while (\$row=mysql_fetch_array(\$query)){
10	\$item=array();
11	\$item["No"]=\$row["no"];
12	\$item["id_raspberry"]=\$row["id_raspberry"];
13	\$item["id_pendaki"]=\$row["id_pendaki"];
14	\$item["waktu"]=\$row["waktu"];
15	\$item["status"]=\$row["status"];
16	mysql_query("update server_pos set
17	status_pengiriman='1'
18	where id_pendaki='".\$row["id_pendaki"]."' and
19	id_raspberry='".\$row["id_raspberry"]."'");
20	array_push(\$respon,\$item);
	}
21	\$response=array();
22	\$response["status"]=true;
23	\$response["data"]=\$respon;
24	}else{
25	
26	\$response=array();
27	
28	\$response["status"]=false;
29	\$response["data"]=null;
30	}

31	<code>echo json_encode(\$response);</code>
32	<code>?></code>

Pada Tabel 5.10 merupakan program yang dijalankan ketika terdapat pendaki baru setelah melakukan pendaftaran di kantor dan kemudian menjalankan aplikasi pendakian. Aplikasi ini akan secara *background service* mengambil data dari server kantor yang telah disediakan untuk diberikan kepada setiap pos yang akan dilewati pendaki. Adapun penjelasan kode program adalah sebagai berikut :

- Baris 2 yakni mengambil file koneksi.php.
- Baris 4 – 5 yaitu mengecek apakah terdapat data di dalam server kantor dengan status pengiriman bernilai 0.
- Baris 7 – 15 jika data \$query lebih dari 0 maka mendeklarasikan variabel \$respon sebagai array kemudian menjalankan perulangan memanggil data dengan diawali mendeklarasikan variabel \$item sebagai array kemudian.
- Baris 16 – 19 mengupdate status_pengiriman milik data yang telah diambil menjadi bernilai 1.
- Baris 20 memanipulasi array dengan elemen \$respon dan \$item.
- Baris 21 mendeklarasikan variabel \$response.
- Baris 22 – 23 mengolah data kemudian dimasukkan ke dalam array bersarang. Jika data status bernilai true maka akan dilanjutkan dengan memasukkan data \$respon ke dalam array.
- Baris 24 – 29 jika tidak maka mendeklarasikan \$response kemudian dilanjutkan dengan mengecek apakah terdapat data status, jika tidak maka tidak ada data yang dimasukkan ke dalam array.
- Baris 31 normalisasi data menjadi format JSON.

5.4.5 Implementasi Pendaki mengupdate status pengiriman di server pos

Pada implementasi pendaki mengupdate status pengiriman di server pos merupakan skenario dimana pendaki yang sudah memulai pendakian dengan membawa data titipan dari server kantor akan dikirimkan kepada setiap server pos yang dia lalui. Data yang dibawa pendaki bertujuan untuk ACK dimana memberitahukan kepada setiap pos bahwa data yang dia titipkan sudah sampai pada server kantor. Data yang dicari pada tahap ini adalah data dengan status pengiriman bernilai 1, yakni data yang tadinya sudah dibawa turun oleh pendaki yang lain. Pendaki yang baru naik ini akan mengubah status pengiriman tersebut menjadi bernilai 2 yang berarti bahwa data tersebut telah sampai pada server kantor. Potongan program dapat dilihat pada Tabel 5.11.

Tabel 5.11 Program UpdateStatus.php

No	File : UpdateStatus.php
1	<code><?php</code>



2	<code>include "koneksi.php";</code>
3	
4	<code>\$id_pendaki=\$_POST ['id_pendaki'];</code>
5	<code>\$query =mysql_query (" update server_pos set</code>
6	<code>status_pengiriman='2' where id_pendaki='\$id_pendaki');</code>
7	
8	<code>if (\$query){</code>
9	<code>echo true;</code>
10	<code>} else {</code>
11	<code>echo false;</code>
12	<code>}</code>
13	<code>?></code>

Pada Tabel 5.11 merupakan program yang dijalankan ketika pendaki yang sedang naik berada pada area pos dan sedang membawa data yang dititipkan oleh server kantor untuk diberikan kepada server pos. Maka secara *background service* aplikasi pendaki akan melakukan update nilai status_pengiriman data di dalam *database* server pos sesuai dengan data yang dibawa oleh pendaki. Adapun penjelasan kode program adalah sebagai berikut :

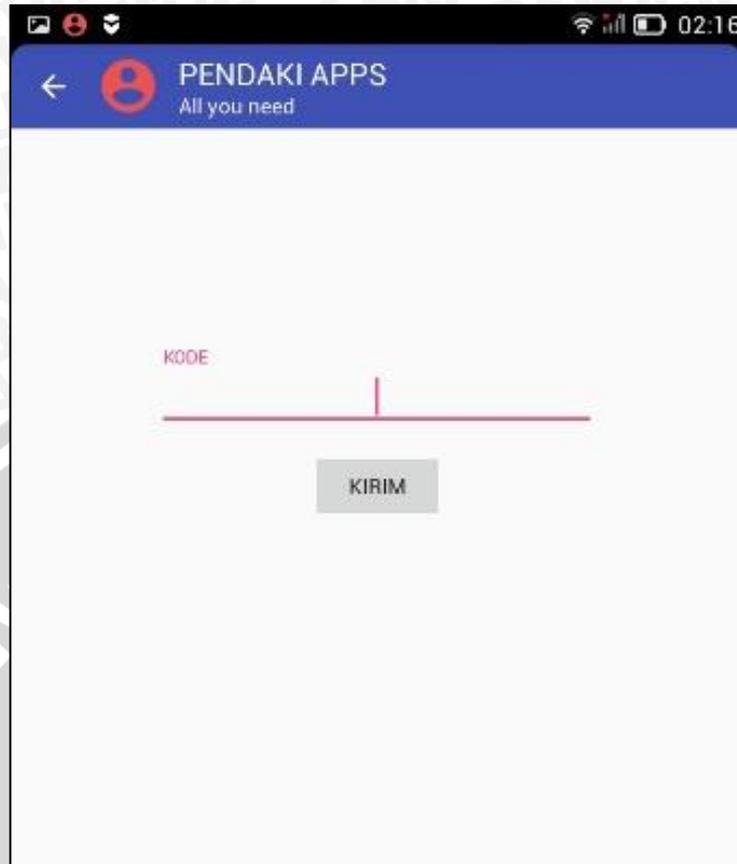
- Baris 2 memanggil koneksi.php.
- Baris 4 mendeklarasikan variabel \$id_pendaki.
- Baris 5 – 6 untuk mengupdate data di dalam *database*. Mengupdate status pengiriman menjadi bernilai 2 jika terdapat ID pendaki sesuai dengan data yang dibawa dari server kantor.

5.5 Implementasi Antarmuka

Pada sub bab ini akan menjelaskan implementasi antarmuka yang telah dibuat, antara lain yaitu aplikasi android sebagai perantara pengiriman dan tampilan keberadaan pendaki berbasis web.

5.5.1 Aplikasi Android

Aplikasi android disini membuat sebuah tampilan utama sederhana yakni untuk memasukkan ID pendaki dan kemudian dikirmkan ke setiap pos. Adapun tampilan dapat dilihat pada Gambar 5.2.



Gambar 5.2 Tampilan utama aplikasi android

Aplikasi android memang dibuat sederhana untuk tampilannya karena pada dasarnya pendaki hanya membutuhkan memasukkan ID pendaki saat pertama aplikasi dijalankan. Untuk selanjutnya semua perlakuan akan dilakukan di dalam *background service*. Adapun potongan program *background service* dapat dilihat pada Tabel

Tabel 5.12 Program *background service* android

No	File : TestService.java
1	<code>public void run() {</code>
2	<code> mHandler.post(new Runnable() {</code>
3	<code> @Override</code>
4	<code> public void run() {</code>
5	
6	<code> syncServerBawah();</code>
7	
8	<code> String message = "PENDAKI SMART";</code>
9	<code> showForegroundNotification(getIdPendaki());</code>
10	<code> if (getIdPendaki().equalsIgnoreCase("SILAKAN ISI ID</code>
11	<code> PENDAKI")) {</code>
12	<code> getDataBawah();</code>



13	<code>}else{</code>
14	<code>Config config = new Config();</code>
15	<code>String pos = SecurePrefManager.with(getContext())</code>
16	<code>.get("pos")</code>
17	<code>.defaultValue("1")</code>
18	<code>.go();</code>
19	
20	<code>if(getPos(String.valueOf(config.maxPos())) < 1){</code>
21	<code> syncIdPendaki();</code>
22	<code> getDataTitip();</code>
23	<code> message = "BELUM SAMPAI PUNCAK, sekarang pos : " + pos;</code>
24	<code> }else if(pos.equalsIgnoreCase("0")) {</code>
25	<code> syncServerBawah();</code>
26	<code> }else</code>
27	<code> if(pos.equalsIgnoreCase(String.valueOf(config.maxPos()))</code>
28	<code> {</code>
29	<code> syncIdPendaki();</code>
30	<code> syncIdPendakiTurun();</code>
31	<code> getDataTurun();</code>
32	<code> }else{</code>
33	<code> syncIdPendakiTurun();</code>
34	<code> getDataTurun();</code>
35	<code> message = "SUDAH SAMPAI PUNCAK, sekarang pos : " + pos;</code>
36	<code> }</code>
37	<code>}</code>
38	<code>displayNotification(message);</code>
39	<code>}</code>
40	<code>});</code>
41	<code>}</code>

Pada Tabel 5.12 merupakan potongan program dalam aplikasi android dimana langkah berjalannya *background service*. Dalam proses ini akan memanggil perintah yang berada pada web service. Adapun penjelasan program adalah sebagai berikut :

- Baris 4 - 6 program diawali dari pendaki baru yang mengambil data titip dari server kantor tanpa diketahui oleh pendaki.
- Baris 8 – 9 aplikasi akan menampilkan string message serta menampilkan id milik pendaki.
- Baris 10 – 18 proses jika ID pendaki baru belum dimasukkan maka akan melakukan pengambilan data yang siap ditiptikan oleh server kantor.

- Baris 20 – 23 jika pos dimana pendaki berada kurang dari jumlah pos puncak maka akan melakukan proses pendaki memasukkan ID miliknya dan kemudian aplikasi akan secara *background service* melakukan proses update ACK sesuai dengan skenario mengupdate status pengiriman pada data di dalam *database* server pos sesuai dengan data yang dibawa dari server kantor.
- Baris 24 – 25 jika pos dimana pendaki berada merupakan pos 0 atau didefinisikan sebagai server kantor maka proses yang dilakukan adalah aplikasi akan secara *background service* memasukkan data yang telah dibawa dari server pos.
- Baris 26 – 31 jika pos dimana pendaki berada merupakan pos puncak maka yang dilakukan adalah aplikasi pendaki secara *background service* memasukkan ID dengan status “naik” kemudian dilanjutkan dengan memasukkan ID dengan status “turun” dan selanjutnya mengambil data dari server pos yang siap dititipkan untuk ditujukan ke server kantor.
- Baris 32 – 35 kondisi lain yakni dimana pendaki turun telah melewati pos puncak dan melakukan perjalanan turun maka aplikasi akan secara *background service* memasukkan ID dengan status “turun” dan kemudian mengambil data dari server pos yang siap dititipkan untuk ditujukan ke server kantor.

5.5.2 Tampilan Keberadaan Pendaki

Tampilan keberadaan pendaki dibuat dengan berbasis web. Dimana tampilan ini berdasarkan *database* server kantor. Setiap 10 menit tampilan ini akan dibuat update data atau refresh untuk mengetahui apakah ada data yang baru masuk ke dalam *database* server kantor dan butuh untuk ditampilkan. Tampilan ini sangat berguna bagi petugas untuk memantau setiap pendaki yang sedang melaksanakan perjalanannya. Berikut potongan program dapat dilihat pada Tabel 5.13.

Tabel 5.13 Program Tampilan *Database* Kantor

No	frametampilan.php
1	<!DOCTYPE html>
2	<html>
3	<head>
4	<title></title>
5	</head>
6	<body>
7	<iframe src="tampilan1.php" id="tampilan1" height="500"
8	width="600"></iframe>
9	<iframe src="tampilan2.php" id="tampilan2" height="500"
10	width="600"></iframe>

11	<iframe src="tampilan3.php" id="tampilan3" height="500"
12	width="600"></iframe>
13	<script type="text/javascript">
14	window.setInterval(function(){reloadIFrame();},300000);
15	function reloadIFrame(){
16	console.log("reload");
17	
18	var tamp1 = document.getElementById('tampilan1');
19	tamp1.src = tampil1.src;
20	var tamp2 = document.getElementById('tampilan2');
21	tamp2.src = tampil2.src;
22	var tamp3 = document.getElementById('tampilan3');
23	tamp3.src = tampil2.src;
24	}
25	</script>
26	</body>
27	</html>

Pada Tabel 5.13 merupakan program untuk menampilkan isi dari *database* server kantor yang bertujuan agar dapat digunakan petugas kantor untuk melakukan pemantauan terhadap keberadaan pendaki. Adapun penjelasan kode program adalah sebagai berikut :

- Baris 14 merupakan waktu reload yang akan dilakukan oleh program.
- Baris 15 – 24 adalah proses memanggil file tampilan1.php, tampilan2.php, dan tampilan3.php.

Potongan kode tersebut akan menampilkan data keberadaan ID pendaki berdasarkan nama posnya. Pada tampilan ini akan terlihat setiap pos telah dilalui oleh pendaki siapa saja, maka jika terjadi kasus hilangnya pendaki akan dapat diketahui keberadaan pendaki saat terakhir sebelum dinyatakan hilang.

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini akan menjelaskan hasil dari pengujian dalam implementasi metode *store and forward* pada sistem pemantauan pendaki gunung. Pengujian ini dilakukan untuk mengetahui keberhasilan sistem dalam mengimplementasikan skenario yang dirancang. Pada hal ini penulis akan melakukan pengujian secara kuantitatif, dimana setiap tahap skenario implementasi diuji dan mengetahui hasilnya apakah masih terdapat kesalahan. Penulis akan melakukan dua pengujian yakni pengujian konektivitas dan pengujian implementasi *store and forward*. Adapun pengujian konektivitas bertujuan untuk mengetahui hasil dari upaya penulis untuk membangun sebuah koneksi lokal di daerah pegunungan dengan cara membangun *access point* pada setiap pos dan kantor. Ini bertujuan agar pendaki dapat selalu terpantau keberadaannya. Sedangkan pengujian implementasi *store and forward* bertujuan untuk mengetahui apakah pengiriman yang data yang dilakukan dengan bantuan pendaki dapat berjalan dengan lancar. Pengiriman data ini dapat dilakukan dari dua arah, yakni dari kantor yang menitipkan data kepada pendaki untuk dibawa naik dan disampaikan pada setiap pos, arah yang berbeda dilakukan dari pos yang menitipkan data kepada pendaki yang turun untuk disampaikan kepada server kantor.

6.1 Pengujian Konektivitas

Pada pengujian konektivitas yakni menguji apakah *access point* yang dibangun telah benar dapat digunakan oleh pendaki. Adapun skenario pengujiannya adalah awalnya dengan mengaktifkan *access point* pada *Raspberry pi* dan kemudian pendaki menyambungkan wifi smartphone miliknya dengan wifi lokal yang disediakan. Pendaki yang baru berada di wilayah pegunungan akan secara langsung diarahkan untuk menyambungkan wifi miliknya pada pos pendaftaran. Pada tahap ini merupakan tahap awal yang sangat penting untuk dapat melaksanakan skenario selanjutnya yakni pengiriman data menggunakan metode *store and forward*. Smartphone milik pendaki akan menjadi salah satu titik utama dalam pengiriman data ketika sudah tersambung dengan wifi lokal yang disediakan serta sudah diinstall aplikasi pendakian.

Access point dibangun pada semua *Raspberry pi* atau di setiap pos dan kantor dengan menggunakan SSID dan ip yang sama. Tujuannya yakni jika terdapat smartphone pendaki yang sudah terhubung pada *access point* kantor maka untuk selanjutnya di setiap pos akan secara otomatis menyambungkan dirinya.

Pada pengujian konektivitas mempunyai beberapa parameter keberhasilan yaitu :

- *Raspberry pi* dapat mengaktifkan *access point* secara langsung setelah dinyalakan.
- Smartphone pendaki dapat mendeteksi wifi lokal dari *Raspberry pi*.

- Smartphone dapat terkoneksi dengan wifi lokal dari *Raspberry pi*.

Dalam melakukan pengujian konektivitas antara smartphone dengan *Raspberry pi* maka beberapa skenario untuk menguji seperti yang sudah dijelaskan diatas, berikut hasil dari setiap skenario antara lain :

- **Skenario 1**

Menghidupkan access poin dalam *Raspberry pi*.

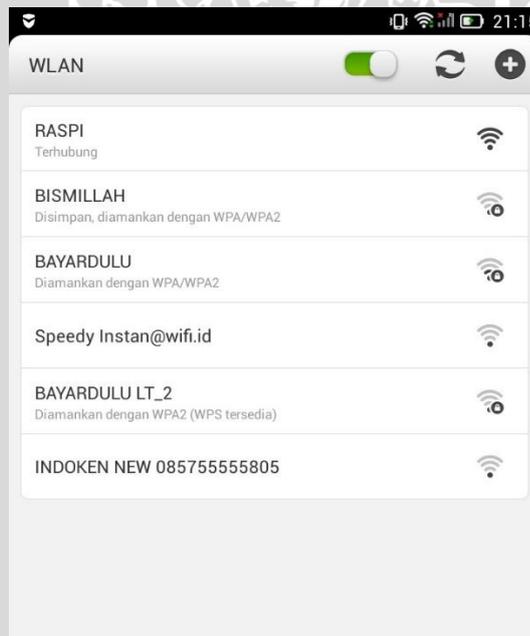
```
pi@raspberrypi:~$ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Using interface wlan0 with hwaddr ac:a2:13:62:01:e7 and ssid "RASPI"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
```

Gambar 6.1 Tampilan *access point* enable

Pada Gambar 6.1 dapat dilihat bahwa nama SSID dari *access point* ini yaitu RASPI, dan mengizinkan untuk digunakan. Ketika itu berarti *access point* sudah dapat terdeteksi pada smartphone pendaki dan pendaki dapat menyambungkan wifi miliknya.

- **Skenario 2**

Smartphone dapat mendeteksi dan terhubung pada *access point*.



Gambar 6.2 Tampilan pada smartphone

Pada Gambar 6.2 dapat dilihat bahwa *access point* dengan nama RASPI sudah dapat terdeteksi oleh smartphone. Selanjutnya smartphone dapat terhubung ke *Raspberry pi* di setiap pos dengan otomatis karena setiap pos mempunyai *access point* dengan nama dan ip yang sama.

- Skenario 3

Raspberry pi menampilkan bahwa terdapat smartphone yang terhubung dengan *access point*.

```
pi@raspberrypi:~$ sudo /usr/sbin/hostapd /etc/hostapd/hostapd.conf
Configuration file: /etc/hostapd/hostapd.conf
Using interface wlan0 with hwaddr ac:a2:13:62:01:e7 and ssid "RASPI"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
wlan0: STA cc:07:e4:7e:4c:58 IEEE 802.11: authenticated
wlan0: STA cc:07:e4:7e:4c:58 IEEE 802.11: associated (aid 1)
wlan0: AP-STA-CONNECTED cc:07:e4:7e:4c:58
wlan0: STA cc:07:e4:7e:4c:58 RADIUS: starting accounting session 5793DF64-000000
00
```

Gambar 6.3 Tampilan *access point* sedang digunakan

Pada Gambar 6.3 dapat dilihat bahwa terdapat satu smartphone yang berhasil terhubung dengan *access point* ini. Yakni smartphone yang mempunyai mac address cc:07:e4:7e:4c:58. Dari hasil pengujian diatas membuktikan bahwa *access point* yang dibangun dapat digunakan dan smartphone juga dapat dengan mudah terhubung pada wifi ini.

6.2 Pengujian Implementasi *Store and forward*

Pada tahap pengujian ini akan mengetahui keberhasilan sistem dalam mengimplementasikan sesuai dengan skenario. Pengujian implementasi *store and forward* akan menampilkan setiap perlakuan antara pendaki dengan server pos begitu pula pendaki dengan server kantor. Adapun parameter keberhasilan yang harus dicapai saat melakukan pengujian antara lain :

- Pendaki hanya melakukan insert ID ketika awal menjalankan aplikasi pendakian, selanjutnya melakukan insert ID dengan status “naik” secara *background service*
- Pendaki melakukan insert ID dengan status “turun” secara *background service*.
- Pendaki dapat secara *background service* mengambil data dari server pos.
- Pendaki dapat secara *background service* memasukkan data ke server kantor.
- Pendaki dapat secara *background service* mengambil data dari server kantor.
- Pendaki dapat secara *background service* mengupdate status pengiriman data pada *database* server pos.

Dalam melakukan pengujian implementasi *store and forward* maka dilakukan beberapa skenario antara lain :

- Skenario 1

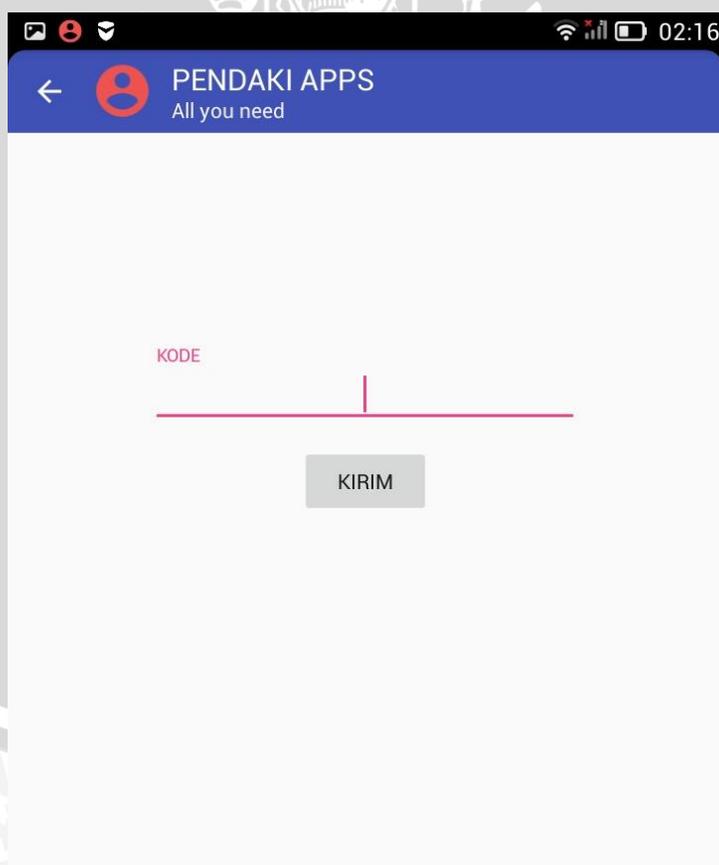
Target yang dituju yakni

- Pendaki dapat mengambil data dari server kantor secara

background service.

- Status pengiriman dari data yang diambil berubah menjadi bernilai 1

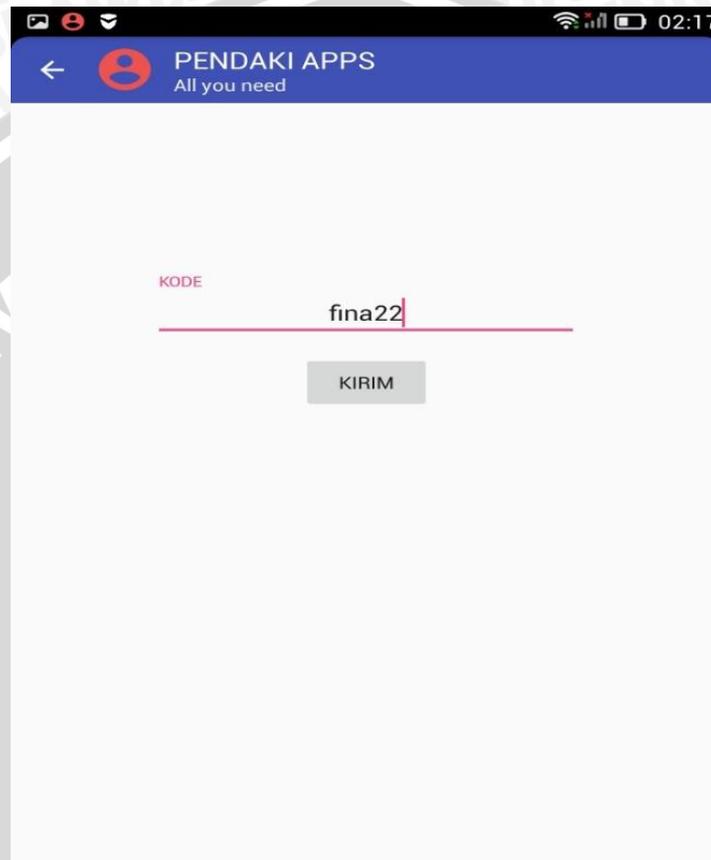
Adapun skenario ini dilakukan oleh pendaki baru yang akan melakukan pendakian dan kemudian pendaki tersebut melakukan pendaftaran pada pos pendaftaran. Saat melakukan pendaftaran, pendaki akan mendapatkan ID dari petugas. Pendaki akan diarahkan untuk menginstall aplikasi pendakian di dalam smartphone miliknya. Setelah aplikasi diinstall maka selanjutnya pendaki akan diarahkan untuk menyambungkan wifi dengan *access point* RASPI yang telah disediakan. Setelah smartphone berhasil dijalankan maka selanjutnya pendaki dapat menjalankan aplikasi pendakian yang telah diinstall. Kemudian pendaki akan dapat memasukkan ID miliknya di dalam aplikasi tersebut. Dan ketika itu juga aplikasi akan secara *background service* mengambil data ACK yang akan dititipkan oleh server kantor untuk dibawa ke server pos.

The image shows a screenshot of an Android application interface. At the top, there is a blue header bar with a back arrow on the left, a red circular icon with a white person silhouette in the center, and the text "PENDAKI APPS" and "All you need" on the right. Below the header, the main content area is white. It features a label "KODE" in red text above a horizontal red line representing an input field. Below the input field, there is a grey button with the text "KIRIM" in white.

Gambar 6.4 Tampilan form insert ID pendaki

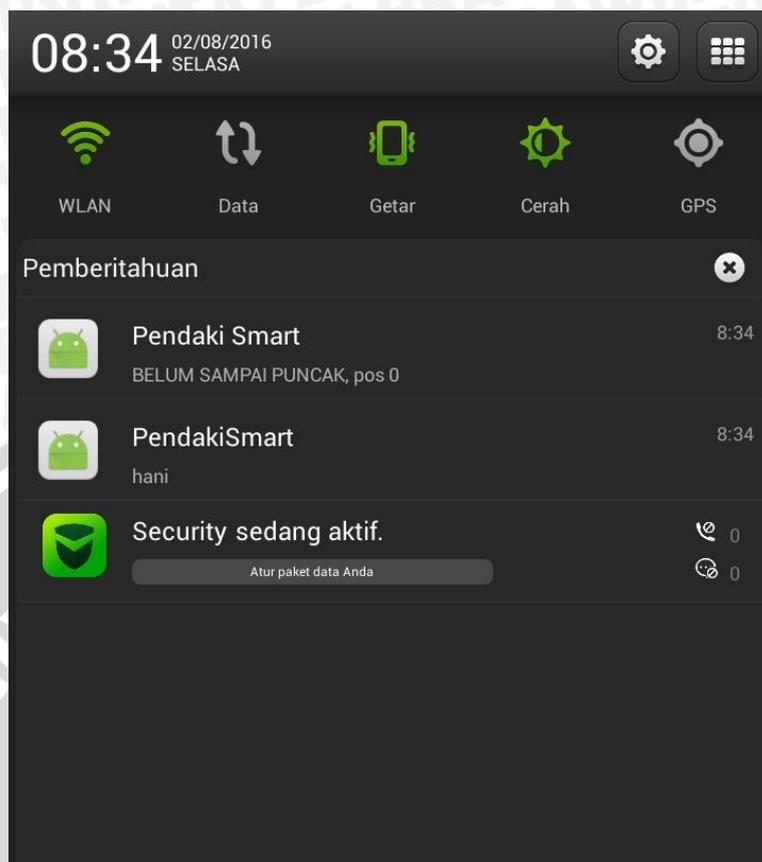
Pada Gambar 6.4 merupakan tampilan form pada aplikasi android dimana pendaki harus memasukkan ID yang dia miliki. ID tersebut

didapatkan ketika pendaki melakukan pendaftaran. Dapat dilihat dari tampilan sudah sesuai dengan desain antarmuka yang telah dijelaskan pada bab perancangan. Penulis membuat aplikasi android dengan desain antarmuka yang sederhana bertujuan agar pendaki hanya akan menjalankan aplikasi satu kali di awal tanpa harus mengganggu perjalanan.



Gambar 6.5 Tampilan Pendaki memasukkan id

Pada Gambar 6.5 merupakan tampilan dimana pendaki memasukkan ID yang dimiliki ke dalam kolom yang telah disediakan. Adapun proses yang dilakukan pendaki dalam memasukkan ID yakni dengan menuliskannya di kolom yang telah disediakan dan selanjutnya pendaki menekan tombol "kirim". Kemudian setelah itu sudah dilakukan maka aplikasi dapat ditutup dan aplikasi akan berjalan secara *background service*. ID yang diinputkan akan terlebih dahulu tersimpan di dalam *database android*.



Gambar 6.6 Tampilan *Background service*

Adapun tampilan dari keberhasilan insert tersebut akan ditampilkan pada Gambar 6.6. Pada gambar menunjukkan bahwa ID “hani” telah berhasil masuk ke dalam *database* dan siap untuk diforward pada setiap pos yang akan dilewati secara background server. Adapun wilayah dimana pendaki sekarang berada dapat dilihat pada keterangan “BELUM SAMPAI PUNCAK, pos 0”. Saat ini pendaki masih dalam wilayah kantor belum sampai pada wilayah pos pertama. Jika telah berada pada wilayah pos maka akan ditampilkan pos berapa sekarang pendaki berada.



Gambar 6.7 Tampilan *Database* Di dalam Server Kantor

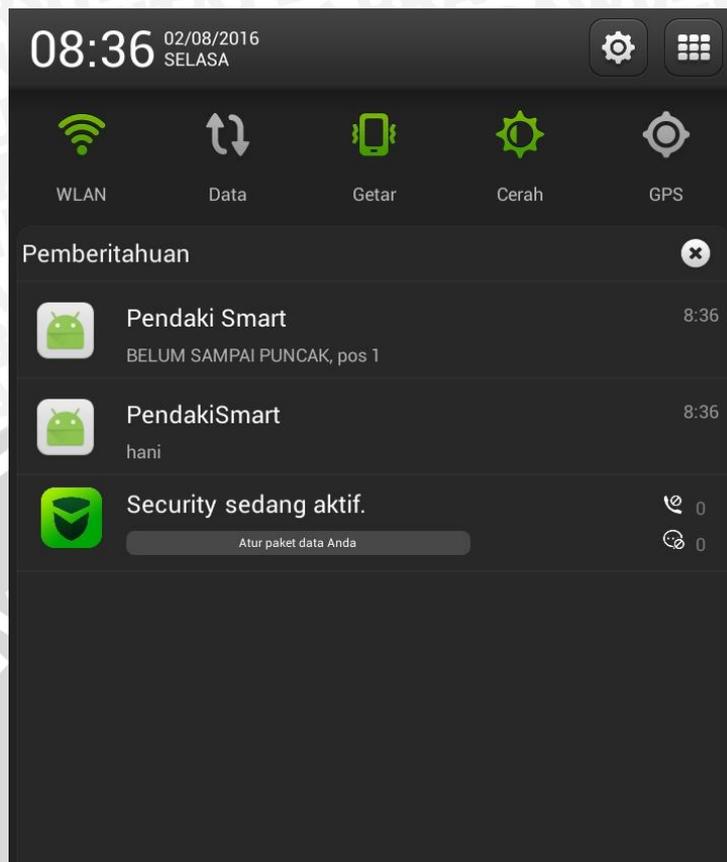
Gambar 6.7 merupakan tampilan *database* dari server kantor. Dimana *database* sedang tidak memiliki data yang siap untuk dititipkan yang berarti bahwa pendaki akan mendaki dengan tidak membawa data ACK melainkan hanya membawa data milik dirinya sendiri yakni ID pendaki. Dari hasil pengujian diatas maka dapat dinyatakan bahwa skenario 1 telah berhasil mencapai target yang dituju.

- Skenario 2

Target yang dituju yakni

- Pendaki dapat memasukkan ID dengan status “naik” secara *background service*.
- Status pengiriman data milik pendaki yang baru dimasukkan akan bernilai 0.
- Pendaki secara *background service* melakukan update status pengiriman data di dalam server pos yang sebelumnya bernilai 1 menjadi bernilai 2.

Adapun skenario ini dilakukan oleh pendaki yang sedang melakukan pendakian dan telah melakukan tahap – tahap pada skenario 1. Pendaki diharuskan tetap menghidupkan smartphone untuk dapat melakukan skenario ini. Di dalam skenario ini pendaki yang sedang dalam melakukan pendakian akan memasukkan ID dengan status “naik” secara *background service* ketika terhubung dengan *access point* RASPI yang berada di setiap pos. Ketika pendaki memasukkan data ID dan status miliknya ke dalam server pos maka data tersebut akan memiliki status pengiriman bernilai 0. Pendaki akan menyampaikan data yang telah dititipkan oleh server kantor untuk disampaikan kepada server pos dengan mengupdate data di dalam server pos yang memiliki status pengiriman bernilai 1 menjadi bernilai 2 sesuai data yang dibawa pendaki.



Gambar 6.8 Tampilan *Background service* – pos 1

Pada Gambar 6.8 menunjukkan bahwa pendaki telah sampai dalam area pos 1. Ketika pendaki berada di area pos 1, maka aplikasi akan secara *background service* memasukkan ID dengan status “naik”. Di smartphone akan menampilkan keberadaan pos setelah aplikasi berhasil memasukkan ID beserta statusnya. Pada tampilan *background service* juga akan ditampilkan ID milik pendaki.

Urut berdasarkan kunci: Tidak ada

+ Opsi		No	id_raspberry	id_pendaki	waktu	status	status_pengiriman
<input type="checkbox"/>	Ubah Salin Hapus	302	1	gina	2016-07-24 05:49:07	naik	1
<input type="checkbox"/>	Ubah Salin Hapus	307	1	gina	2016-07-24 05:49:07	turun	1
<input type="checkbox"/>	Ubah Salin Hapus	308	1	fina22	2016-07-24 05:37:18	naik	0
<input type="checkbox"/>	Ubah Salin Hapus	309	1	hani	2016-08-02 07:31:53	naik	0

Gambar 6.9 Tampilan *Database server pos 1*

Untuk memastikan apakah data pendaki “hani” telah masuk ke dalam *database server pos* maka kita dapat melihat isi dalam *database server pos 1*. Sesuai dengan Gambar 6.9 maka menunjukkan bahwa data “hani” telah masuk ke dalam *database server pos 1*. Sesuai dengan target yakni data yang dimasukkan oleh pendaki “hani” dengan status “naik”

memiliki status pengiriman bernilai 0. Status pengiriman bernilai 0 dikarenakan data ini merupakan data baru yang belum pernah dititipkan kepada pendaki. Data yang sudah pernah dititipkan kepada pendaki maka memiliki status pengiriman bernilai 1.

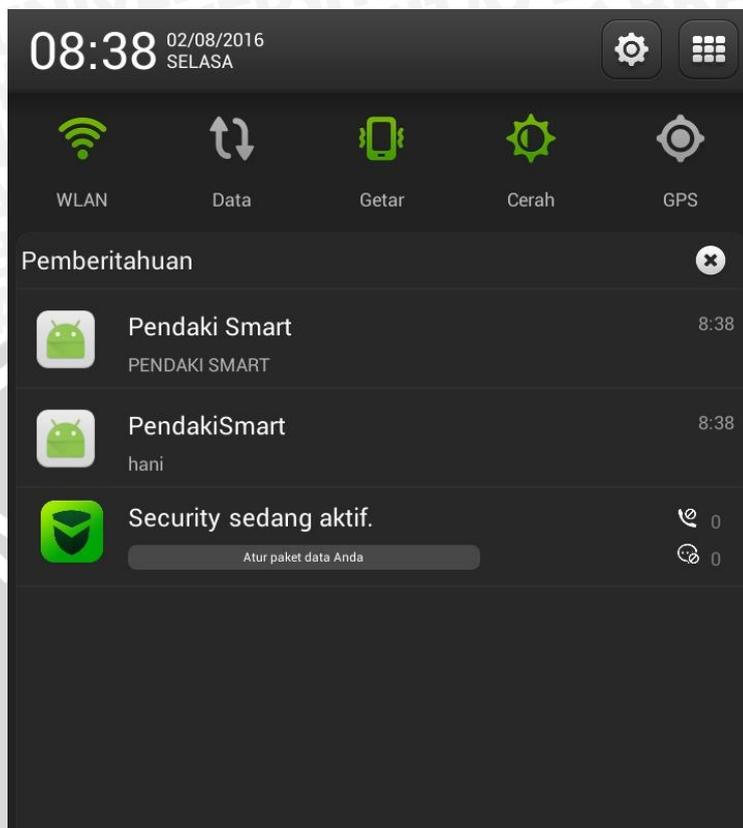
Pada Gambar 6.9 menunjukkan bahwa pendaki “hani” tidak melakukan update *database* untuk data yang memiliki status pengiriman bernilai 1 menjadi data dengan status pengiriman bernilai 2. Ini dikarenakan pendaki “hani” ketika berada pada area kantor tidak mengambil data apapun atau kantor tidak menitipkan data karena *database* masih dalam kondisi kosong. Dari hasil pengujian diatas maka dapat dinyatakan bahwa skenario 2 telah berhasil mencapai target yang dituju.

- Skenario 3

Target yang dituju yakni

- Pendaki dapat memasukkan ID dengan status “naik” secara *background service*.
- Pendaki dapat memasukkan ID dengan status “turun” secara *background service*.
- Status pengiriman data milik pendaki yang baru dimasukkan akan bernilai 0.
- Pendaki secara *background service* mengambil data yang sudah disiapkan server pos untuk dikirimkan ke server kantor.
- Pendaki secara *background service* mengupdate status pengiriman data yang telah diambil menjadi bernilai 1.

Adapun skenario ini dilakukan oleh pendaki yang telah mencapai pos puncak dan sedang menuruni pegunungan. Pendaki diharuskan tetap menghidupkan smartphone miliknya. Di dalam skenario ini pendaki dapat secara *background service* memasukkan ID dengan status “turun” ketika smartphone telah terhubung dengan *access point* RASPI. Ketika data yang dimasukkan oleh pendaki masuk ke dalam server pos maka data tersebut akan memiliki status pengiriman bernilai 0. Dalam waktu yang bersamaan pendaki akan secara *background service* mengambil data yang telah disiapkan oleh server pos untuk diserahkan kepada server kantor. Data yang siap diambil yakni data di dalam server pos yang memiliki status pengiriman bernilai 0. Kemudian status pengiriman data akan berubah menjadi bernilai 1 ketika telah diambil oleh pendaki. Skenario ini hanya dapat dilakukan oleh pendaki yang telah melalui pos puncak. Jika pendaki belum melalui pos puncak maka skenario ini tidak dapat dijalankan.



Gambar 6.10 Tampilan *Background service* – pos 2

Pada Gambar 6.10 menunjukkan bahwa pendaki “hani” telah memasuki area pos puncak, yakni pos 2. Gambar menunjukkan bahwa pendaki “hani” telah memasukkan ID dengan status “naik” ke dalam *database* server pos. Jika sudah berada pada pos puncak maka aplikasi akan secara langsung melanjutkan insert ID dengan status “turun”. Adapun isi tampilan *database* dapat dilihat pada Gambar 6.11.

<input type="checkbox"/>				308	2	fina22	2016-08-02 07:18:50	naik	1
<input type="checkbox"/>				309	2	hani	2016-08-02 07:18:50	naik	1
<input type="checkbox"/>				310	2	hani	2016-08-02 07:18:50	turun	1

↑ Pilih Semua Dengan pilihan: Ubah Hapus Ekspor

Gambar 6.11 Tampilan *Database* server pos 2

Pada Gambar 6.11 menampilkan isi *database* server pos 2 dimana dapat dilihat data pendaki “hani” akan secara langsung memasukkan dua buah data yakni ID dengan status “naik” dan ID dengan status “turun”. Serta pada gambar ditampilkan bahwa status pengiriman bernilai 1, hal ini dikarenakan aplikasi akan secara langsung mengambil data yang memiliki status pengiriman bernilai 0 untuk dibawa turun ke server kantor dan kemudian data tersebut akan secara langsung update status pengiriman

menjadi bernilai 1 yang menunjukkan bahwa data tersebut telah pernah dititipkan kepada pendaki sehingga tidak dapat diambil lagi. Dari hasil pengujian diatas maka dapat dinyatakan bahwa skenario 3 telah berhasil mencapai target yang dituju.

- Skenario 4

Target yang dituju yakni :

- Pendaki dapat memasukkan data yang telah dititipkan kepadanya ke dalam server kantor secara *background service*.
- Status pengiriman data yang dimasukkan pendaki akan bernilai 0.

Adapun skenario ini dilakukan oleh pendaki turun yang sudah sampai pada area kantor dimana pendaki harus tetap menyalakan smartphone miliknya. Di dalam area ini aplikasi akan secara *background service* memasukkan data yang telah dibawa dari setiap pos yang telah dilewati. Adapun data yang dimasukkan ke dalam server kantor akan masuk ke dalam *database* dengan memiliki status pengiriman bernilai 0. Data yang dimasukkan oleh pendaki akan secara langsung dilakukan penyaringan yakni jika sudah terdapat data yang sama di dalam *database* maka data tersebut tidak akan dimasukkan kembali. Maka dengan begini tidak akan terdapat penumpukan data yang sama.

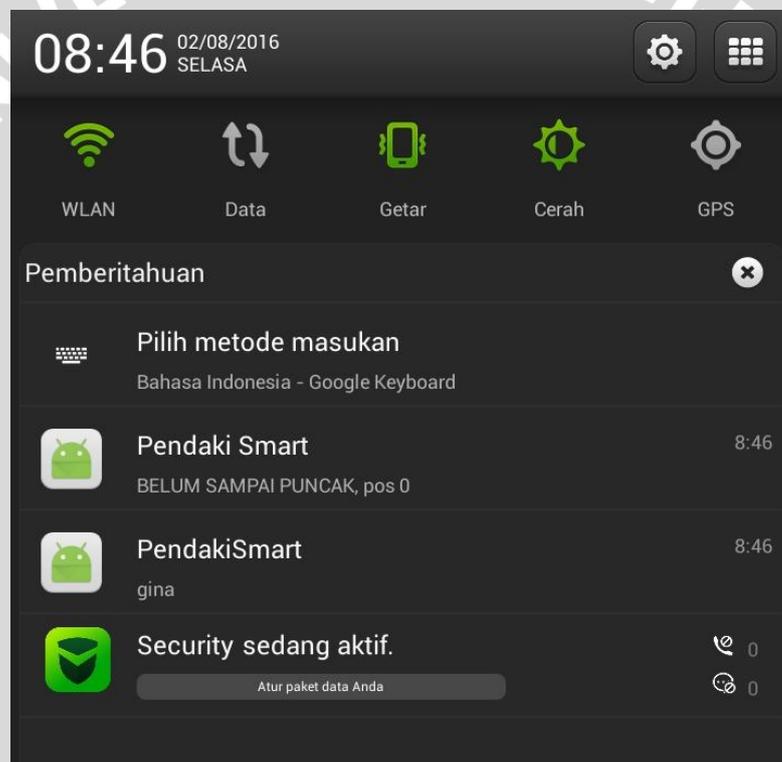
no	id_raspberry	id_pendaki	waktu	status	status_pengiriman
1	2	fina22	2016-07-24 05:41:03	naik	0
2	2	hani	2016-08-02 07:18:40	naik	0
3	2	hani	2016-08-02 07:18:50	turun	0
4	1	fina22	2016-07-24 05:37:18	naik	0
5	1	hani	2016-08-02 07:31:53	naik	0
6	1	hani	2016-08-02 08:41:51	turun	0

Gambar 6.12 Tampilan Isi *Database* Kantor

Pada Gambar 6.12 *database* server kantor telah menerima data yang telah dimasukkan oleh aplikasi pendaki. Data yang dimasukkan akan memiliki status pengiriman bernilai 0. Data yang masuk ke dalam *database* merupakan data yang dibawa dari setiap pos yang dilewati oleh pendaki dan data yang memang disiapkan oleh server pos untuk dititipkan kepada pendaki. Data diatas tidak akan terdapat penumpukan data yang sama karena sebelumnya telah dilakukan pengecekan apakah sudah terdapat data yang sama dengan yang akan dimasukkan di dalam *database*, jika sudah terdapat data yang sama maka data tidak akan dimasukkan kembali. Nilai yang dimiliki oleh status pengiriman yakni bergantung pada kondisi data, jika kondisi data baru masuk dan belum

pernah dititipkan kepada pendaki naik maka status pengiriman akan bernilai 0. Jika data merupakan data lama yang sudah pernah dititipkan kepada pendaki baru untuk disampaikan ke setiap server pos maka data tersebut akan memiliki status pengiriman bernilai 1. Dari hasil pengujian diatas maka dapat dinyatakan bahwa skenario 4 telah berhasil mencapai target yang dituju.

Pengujian selanjutnya akan menunjukkan bahwa terdapat pendaki baru yang melakukan pendaftaran di kantor dan kemudian aplikasi yang dimiliki akan secara *background service* mengambil data di dalam *database* server yang sudah disiapkan untuk disampaikan kepada setiap pos sebagai data ACK. Data yang sudah disiapkan oleh server kantor untuk dibawa oleh pendaki adalah data dengan memiliki status pengiriman bernilai 0.



Gambar 6.13 Tampilan *background service* pendaki “Gina”

Pada Gambar 6.13 menampilkan bahwa terdapat pendaki baru yang telah memasukkan ID di dalam aplikasi pendakian miliknya dan menunjukkan bahwa pendaki sedang dalam area kantor dengan belum melakukan pendakian. Pada area ini maka aplikasi akan secara *background service* melakukan pengambilan data yang sudah disiapkan oleh server kantor untuk dititipkan kepada pendaki dan disampaikan kepada setiap server pos.

Order berdasarkan kolom:

+ Opsi		no	id_raspberry	id_pendaki	waktu	status	status_pengiriman
<input type="checkbox"/>	Ubah Salin Hapus	1	2	fina22	2016-08-02 08:46:30	naik	1
<input type="checkbox"/>	Ubah Salin Hapus	2	2	hani	2016-08-02 08:46:30	naik	1
<input type="checkbox"/>	Ubah Salin Hapus	3	2	hani	2016-08-02 08:46:30	turun	1
<input type="checkbox"/>	Ubah Salin Hapus	4	1	fina22	2016-08-02 08:46:30	naik	1
<input type="checkbox"/>	Ubah Salin Hapus	5	1	hani	2016-08-02 08:46:30	naik	1
<input type="checkbox"/>	Ubah Salin Hapus	6	1	hani	2016-08-02 08:46:30	turun	1

Pilih Semua Dengan pilihan: Ubah Hapus Ekspor

Gambar 6.14 Tampilan Database Kantor Setelah Diambil Pendaki

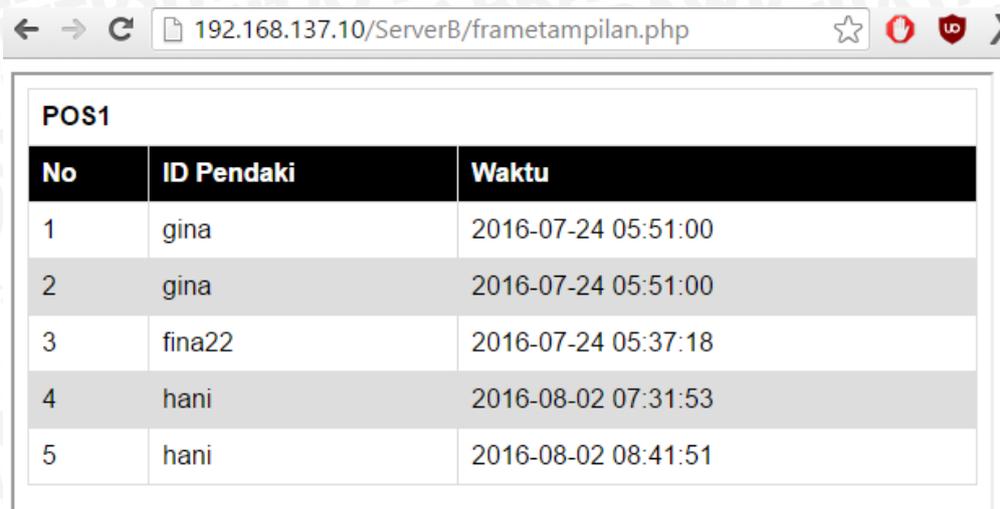
Pada Gambar 6.14 menampilkan isi dari *database* server kantor setelah dilakukan pengambilan oleh aplikasi pendaki secara *background service*. Dapat dilihat bahwa data yang sebelumnya memiliki status pengiriman bernilai 0 sesuai dengan Gambar 6.12 yang berarti bahwa data belum pernah diambil oleh pendaki. Pada Gambar 6.13 menampilkan data yang sama dengan status pengiriman berubah menjadi bernilai 1, yang menunjukkan bahwa data telah diambil oleh pendaki baru yakni pendaki dengan ID “Gina”. Aplikasi pendakian yang dijalankan oleh pendaki telah berhasil mengambil data yang telah disiapkan oleh server kantor untuk disampaikan pada setiap pos yang akan dilewati sebagai data ACK.

- Skenario 5

Target yang dituju yakni :

- Petugas dapat memantau keberadaan pendaki yang ditampilkan dengan berbasis web yang terhubung dengan *database* server kantor.

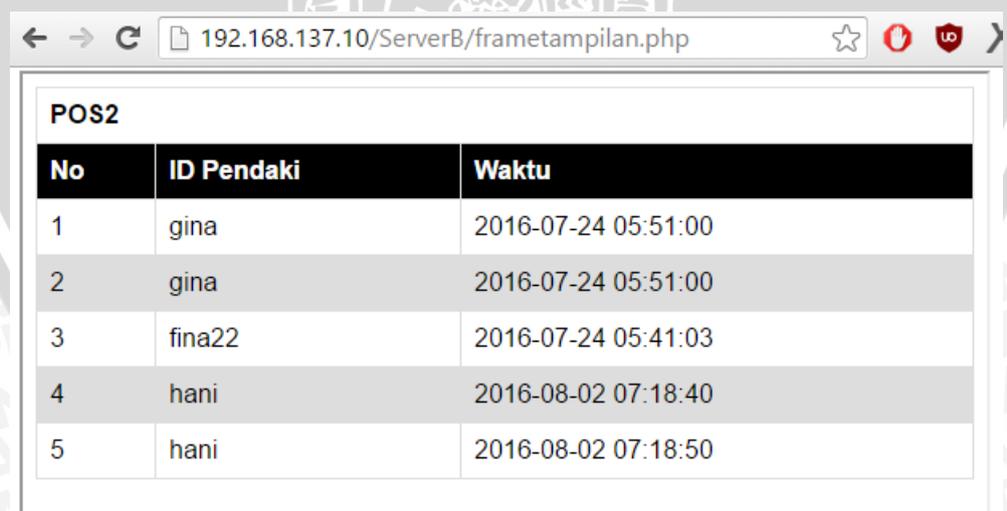
Skenario menjelaskan bahwa terdapat antarmuka yang digunakan untuk petugas dalam melakukan pemantauan pendaki. Antarmuka monitoring pendaki terhubung dengan server kantor yang akan melakukan reload setiap 5 menit. Serta antarmuka ini berbasis web yang ringan untuk dijalankan. Adapun tampilan dari antarmuka akan dapat dilihat pada Gambar 6.15 dan Gambar 6.16.



No	ID Pendaki	Waktu
1	gina	2016-07-24 05:51:00
2	gina	2016-07-24 05:51:00
3	fina22	2016-07-24 05:37:18
4	hani	2016-08-02 07:31:53
5	hani	2016-08-02 08:41:51

Gambar 6.15 Tampilan antarmuka berdasarkan pos 1

Pada Gambar 6.15 merupakan tampilan antarmuka berdasarkan pos 1. Tampilan antarmuka disini disesuaikan dengan isi data pada server kantor. Data yang ada di dalam server kantor akan ditampilkan pada halaman web ini untuk bertujuan memudahkan petugas dalam melakukan pemantauan. Adapun tampilan ini akan melakukan reload setiap 5 menit agar data yang ditampilkan tidak terlalu lama update. Ini juga bertujuan untuk mengetahui apakah terdapat pendaki yang dicurigai membutuhkan bantuan dari petugas. Data ini merupakan bukan data sinkron langsung dengan setiap pos melainkan data yang berada pada server kantor.



No	ID Pendaki	Waktu
1	gina	2016-07-24 05:51:00
2	gina	2016-07-24 05:51:00
3	fina22	2016-07-24 05:41:03
4	hani	2016-08-02 07:18:40
5	hani	2016-08-02 07:18:50

Gambar 6.16 Tampilan antarmuka berdasarkan pos 2

Adapun pada Gambar 6.16 merupakan tampilan antarmuka berdasarkan pos 2. Tampilan antarmuka ini juga merupakan isi dari *database* server kantor yang melakukan reload setiap 5 menit. Data yang ada di dalam server kantor akan secara berkala ditampilkan pada

antarmuka web dan secara mudah diakses oleh para petugas. Dengan tampilan antarmuka ini maka petugas akan dengan mudah melakukan pemantauan terhadap para pendaki. Petugas akan dapat dengan mudah memprediksi jika terdapat pendaki yang dicurigai membutuhkan bantuan atau hilang. Data ini bukan merupakan data sinkron secara langsung dari server pos 2 melainkan sesuai data yang ada di dalam *database* server kantor.

6.3 Analisis Implementasi *Store and Forward*

Pengujian telah dilakukan dengan beberapa skenario yang telah dibuat. Dari semua skenario hasil yang didapatkan adalah sukses mencapai target yang diinginkan. Adapun lebih lengkapnya dapat dilihat pada Tabel 6.1.

Tabel 6.1 Pengujian *Store and forward*

Skenario	Target	Hasil
Skenario 1	Pendaki dapat mengambil data dari server kantor secara <i>background service</i> .	Sukses
	Status pengiriman dari data yang diambil berubah menjadi bernilai 1	Sukses
Skenario 2	Pendaki dapat memasukkan ID dengan status "naik" secara <i>background service</i> .	Sukses
	Status pengiriman data milik pendaki yang baru dimasukkan akan bernilai 0.	Sukses
	Pendaki secara <i>background service</i> melakukan update status pengiriman data di dalam server pos yang sebelumnya bernilai 1 menjadi bernilai 2.	Sukses
Skenario 3	Pendaki dapat memasukkan ID dengan status "naik" secara <i>background service</i> .	Sukses
	Pendaki dapat memasukkan ID dengan status "turun" secara <i>background service</i> .	Sukses
	Status pengiriman data milik pendaki yang baru dimasukkan akan bernilai 0.	Sukses
	Pendaki secara <i>background service</i> mengambil data yang sudah disiapkan server pos untuk dikirimkan ke server kantor.	Sukses
	Pendaki secara <i>background service</i> mengupdate status pengiriman data	Sukses

	yang telah diambil menjadi bernilai 1.	
Skenario 4	Pendaki dapat memasukkan data yang telah dititipkan kepadanya ke dalam server kantor secara <i>background service</i> .	Sukses
	Status pengiriman data yang dimasukkan pendaki akan bernilai 0.	Sukses
Skenario 5	Petugas dapat memantau keberadaan pendaki yang ditampilkan dengan berbasis web yang terhubung dengan <i>database</i> server kantor.	Sukses

Tabel 6.1 menunjukkan hasil pengujian dari skenario beserta target yang telah dibuat. Berdasarkan pengujian yang dilakukan, implementasi pengiriman data menggunakan metode *store and forward* merupakan cara yang baik mengingat kebutuhan untuk dapat memantau keberadaan para pendaki gunung. Dalam mengimplementasikan sistem ini tidak bergantung pada delay maksimum dikarenakan pengiriman data bergantung pada pendaki. Setiap pendaki akan berperan sebagai kurir dengan tujuan mengirimkan data yang telah disiapkan oleh server untuk ditujukan pada destination server. Pengiriman data menggunakan *store and forward* dapat ikut membantu menemukan pendaki yang hilang dengan cepat karena data yang dikirimkan pendaki dari setiap posnya akan masuk pada server kantor dimana petugas akan dengan mudah memantau keberadaan para pendaki.



BAB 7 PENUTUP

Pada bab ini akan dijelaskan kesimpulan yang didapatkan berdasarkan hasil penelitian “Implementasi Metode *Store and Forward* pada Sistem Pemantauan Pendaki Gunung” yang telah dilakukan beserta dengan saran yang akan diberikan untuk penelitian selanjutnya dengan tujuan menyempurnakan penelitian sebelumnya.

7.1 Kesimpulan

Berdasarkan hasil penelitian “Implementasi Metode *Store and Forward* pada Sistem Pemantauan Pendaki Gunung”, dapat disimpulkan bahwa :

- Pembangunan sistem pemantauan pendaki gunung dapat dilakukan dengan membangun access point pada raspberry pi kemudian melakukan pertukaran data lokasi pendaki antara pos dan kantor dengan tujuan mengetahui keberadaan pendaki.
- Pos dan kantor dapat saling bertukar data keberadaan pendaki dengan menggunakan metode *store and forward* dimana pengiriman dilakukan dengan bantuan pendaki. Pos akan mengirimkan data pendaki ke server kantor agar dapat memantau keberadaan pendaki dengan menitipkan kepada pendaki yang sedang turun. Sedangkan kantor akan mengirimkan ACK ke setiap pos untuk mengkonfirmasi bahwa data yang dikirimkan telah diterima dengan cara menitipkan kepada pendaki yang naik.

7.2 Saran

Saran yang dapat diberikan oleh penulis untuk penelitian selanjutnya adalah :

- Untuk penelitian lebih lanjut sistem dapat dikembangkan dengan menambahkan *scheduling* yang lebih tepat untuk proses pengiriman data.
- Sistem dapat dikembangkan dengan menambahkan fitur darurat bagi pendaki yang ingin membutuhkan pertolongan.
- Sistem dapat dikembangkan lagi dengan menambahkan fitur untuk mendapatkan data posisi pendaki yang lebih tepat.

DAFTAR PUSTAKA

- Chosam, L. I., 2015. [Interview] (25 09 2015).
- David Axmark, M. W. P. D., 2001. MySQL Reference Manual for Version 3.23.39.
- Doug Tidwell, J. S. P. K., 2001. *Programming Web Services with SOAP*. s.l.:s.n.
- Fall, K., 2003. A Delay-Tolerant Network Architecture for Challenged Internets.
- Firdaus, F., 2016. *SINDONEWS.com*. [Online] Available at: <http://daerah.sindonews.com/read/1124027/22/nekat-lawan-cuaca-ekstrem-dua-pendaki-hilang-di-gunung-slamet-1468738757> [Accessed 25 03 2016].
- Istiawan, H., 2013. *SINDONEWS.com*. [Online] Available at: <http://daerah.sindonews.com/read/747898/30/menengok-mereka-yang-istirahat-dalam-dekapan-semeru-1370832421> [Accessed 2016].
- Koepfel, A., 2015. Enterprise Smart Outlet: Android Development.
- Lucky, 2008. XML Web services : Aplikasi Desktop, Internet & Handphone.
- Maskov, V., 2015. Implementing REST Client for Android.
- Mitchell, L. J., 2013. PHP Web Services.
- Nazaruddin, S., 2012. Pemrograman Aplikasi Mobile, Smartphone, Dasar..
- Ozcan Koc, J. Y. C.-C. S., 2005. Autonomous Pickup and Delivery for Delay Tolerant Mobile Networks.
- Priyasidharta, D., 2016. *Tempo.co*. [Online] Available at: <https://m.tempo.co/read/news/2016/06/13/058779352/pendaki-asal-swiss-yang-hilang-di-gunung-semeru-masih-misterius> [Accessed 2016].
- Rahmania, L. A., 2013. Penerapan Delay Tolerant Network (DTN) untuk Sistem Konsultasi Kesehatan Jarak Jauh Berbasis Web.
- Sari, N. I., 2015. Implementasi Web Service pada Delay Tolerant Network.
- Siswanti, S. D., 2013. Pengembangan Sistem Aplikasi Pengiriman Data Daerah Terpencil Berbasis Delay Tolerant Network.
- Suriyanto, 2016. *CNN Indonesia*. [Online] Available at: <http://www.cnnindonesia.com/nasional/20160614154501-20-138114/cari-pendaki-swiss-hilang-di-semeru-tim-sar-kerahkan-drone/> [Accessed 29 07 2016].
- Warthman, F., 2015. Delay - and Disruption - Tolerant Networks (DTNs).
- Widhiarti, Y. R., 2014. Digital Repository Materi Pembelajaran Bagi Daerah Tertinggal Berbasis Delay Tolerant Network (DTN).