



**RANCANG BANGUN APLIKASI BERBASIS GUI
(GRAPHIC USER INTERFACE) UNTUK PEMBACAAN
DATA PADA LORA GATEWAY**



Oleh:

Adam Fiqih Ristanto

17410200025

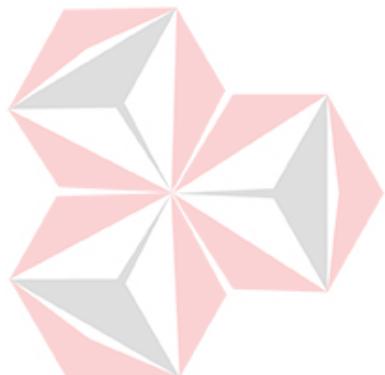
**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2021**

RANCANG BANGUN APLIKASI BERBASIS GUI (*GRAPHIC USER INTERFACE*) UNTUK PEMBACAAN DATA PADA LORA GATEWAY

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan

Program Sarjana Teknik



UNIVERSITAS
Oleh :
Nama : Adam Fiqih Ristanto
NIM : 17410200025
Program : S1 Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2021

TUGAS AKHIR

RANCANG BANGUN APLIKASI BERBASIS GUI (GRAPHIC USER INTERFACE) UNTUK PEMBACAAN DATA PADA LORA GATEWAY

Dipersiapkan dan disusun oleh

Adam Fiqih Ristanto

NIM : 17410200025

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahasan

Pada : 6 Juli 2021

Susunan Dewan Pembahasan

Pembimbing:

- I. Pauladie Susanto, S. Kom., M.T.
NIDN 0729047501
- II. Yosefine Triwidvastuti, M.T.
NIDN 0729038504

Pembahasan:

- I. Dr. Jusak
NIDN 0708017101

Digitally signed by
Universitas Dinamika
Date: 2021.08.04
11:51:53 +07'00'

Digitally signed by
Universitas
Dinamika
Date: 2021.08.04
11:51:53 +07'00'

Digitally signed
by Universitas
Dinamika
Date: 2021.08.04
14:01:17 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar Sarjana

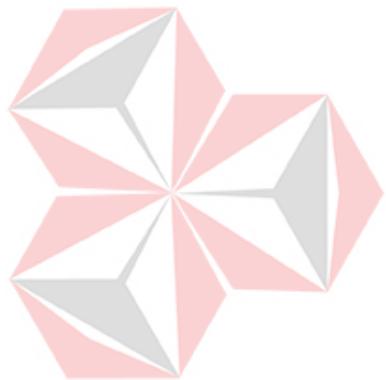
Digitally signed by
Universitas Dinamika
Date: 2021.08.05
09:01:23 +07'00'

Tri Sagirani, S.Kom., M.MT.

NIDN: 0731017601

Dekan Fakultas Teknologi dan Informatika

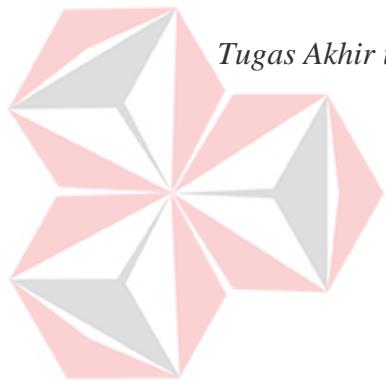
UNIVERSITAS DINAMIKA



“The only source of knowledge is experience.”

- Albert Einstein -

UNIVERSITAS
Dinamika



Tugas Akhir ini kupersembahkan untuk orang tua, saudara, teman-teman, dan guru-guru yang tak pernah lelah menyemangatku.

UNIVERSITAS
Dinamika

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Adam Fiqih Ristanto
NIM : 17410200025
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Kerja : Tugas Akhir
Judul Karya : **RANCANG BANGUN APLIKASI BERBASIS GUI (GRAPHIC USER INTERFACE) UNTUK PEMBACAAN DATA PADA LORA GATEWAY**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyataui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmedikasi dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 22 Juni 2021


Yang Menyatakan,
Adam Fiqih Ristanto
NIM 17.41020.0025

ABSTRAK

Salah satu jaringan yang menggunakan konsep IoT adalah LPWAN (*Low Power Wide Area Network*) dengan perangkatnya yang bernama LoRa. Terdapat perangkat LoRa Gateway berfungsi sebagai pusat masuknya data perangkat LoRa atau node. Untuk membangun sistem keamanan menggunakan LoRa Gateway dibutuhkan aplikasi untuk memantau data node, kemudian dilakukanlah pengujian keakuratan terima tiga data node terhadap LoRa Gateway dengan hasil *loss packet* masing-masing node sebesar 6.04% untuk node 1, 1.13% untuk node 2, dan 1,9% untuk node 3 dimana angka-angka tersebut menunjukkan bahwa aplikasi menerima data node dengan akurat. Setelah itu pengujian kelayakan aplikasi dari sudut pandang pengguna dilakukan dengan menyebar angket yang berisikan 18 pertanyaan kepada 13 orang secara acak dan menghasilkan kesan yang bagus dari para pengguna aplikasi. Sehingga rancang bangun aplikasi berbasis GUI (*Graphic User Interface*) pemantau data node ini dapat memantau dan membedakan ketiga data node sekaligus dengan nilai *packet loss* rendah, serta memenuhi syarat HCI (*Human Computer Interaction*).

Kata Kunci : LoRa Gateway, *loss packet*, node, GUI, HCI

KATA PENGANTAR

Puji dan syukur atas kehadiran Tuhan Yang Maha Esa karena atas rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan tugas akhir yang berjudul “Rancang Bangun Aplikasi Berbasis GUI (*Graphic User Interface*) untuk Pembacaan Data pada LoRa Gateway”. Dalam pelaksanaan Tugas Akhir dan penyelesaian laporan Tugas Akhir ini, penulis mendapatkan bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan rahmatnya dan hidayahnya penulis dapat menyelesaikan Tugas Akhir ini.
2. Orang Tua dan Seluruh Keluarga tercinta yang telah memberikan dorongan dan bantuan baik moral maupun materi sehingga penulis dapat menempuh tugas akhir dan dapat menyusun laporan Tugas Akhir.
3. Bu Tri Sagirani, S.Kom., M.MT. selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
4. Bapak Dr. Jusak selaku Dosen Pembahas atas saran dan masukannya sehingga penulis dapat menyelesaikan laporan Tugas Akhir ini.
5. Bapak Pauladie Susanto, S.Kom., M.T. selaku dosen pembimbing 1 yang telah memberikan saran dan juga wawasan bagi penulis selama pelaksanaan Tugas Akhir dan pembuatan laporan Tugas Akhir.
6. Ibu Yosefine Triwidayastuti, M.T. selaku dosen pembimbing 2 yang memberikan masukan serta koreksi bagi penulis selama pelaksanaan Tugas Akhir dan penyusunan laporan Tugas Akhir.
7. Seluruh dosen pengajar Program Studi S1 Teknik Komputer yang telah mendidik, memberi motivasi kepada penulis selama masa kuliah di Universitas Dinamika.
8. Teman-teman seperjuangan Teknik Komputer angkatan 2017 dan semua pihak yang terlibat namun tidak dapat penulis sebutkan satu persatu atas bantuan dan dukungannya.

Penulis berharap semoga laporan ini dapat berguna dan bermanfaat untuk menambah wawasan bagi pembacanya. Penulis juga menyadari dalam penulisan laporan ini banyak terdapat kekurangan. Oleh karena itu penulis sangat

mengharapkan saran dan kritik untuk memperbaiki kekurangan dan berusaha untuk lebih baik lagi.

Surabaya, 20 Juli 2021

Penulis



DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II LANDASAN TEORI	4
2.1 RHF0M301 LoRa Gateway	4
2.2 Raspberry PI 3 B+.....	6
2.3 GPS (<i>Global Positioning System</i>)	7
2.4 GUI (<i>Graphical User Interface</i>)	7
2.5 Python.....	8
2.5.1 Tkinter.....	8
BAB III METODOLOGI PENELITIAN	9
3.1 Model Perancangan	9
3.1.1 Node.....	9
3.1.2 Receiver.....	9
3.1.3 Mini-PC	10

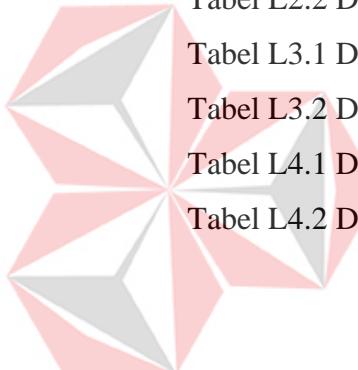
3.1.4 Visualisasi Data	10
3.2 Skematik Perangkat Keras	11
3.2.1 Skematik Gateway	11
3.3 Perancangan GUI	12
3.3.1 Desain GUI	12
3.3.2 Posisi Node Dalam Map	13
3.4 Perancangan Perangkat Lunak	13
3.4.1 Algoritma Sistem Gateway	14
3.4.2 Pengolahan Data Node Menjadi GUI	15
BAB IV HASIL DAN PEMBAHASAN	16
4.1 Aplikasi Pemantau Data Node	16
4.1.1 Program GUI Menggunakan Tkinter	16
4.1.2 LoRa Gateway sebagai receiver	24
4.1.3 Threading	26
4.2 Uji coba keakuratan terima data dan kecepatan waktu proses tampilan	27
4.2.1 Uji coba keakuratan terima data	28
4.2.2 Evaluasi kecepatan waktu proses tampilan.....	29
4.3 Angket pengalaman pengguna setelah mengoperasikan aplikasi	29
BAB V KESIMPULAN DAN SARAN	33
5.1 Kesimpulan.....	33
5.2 Saran	33
DAFTAR PUSTAKA	34
BIODATA PENULIS.....	36
LAMPIRAN.....	37

DAFTAR GAMBAR

	Halaman
Gambar 2.1 RHF0M301 LoRa Gateway	4
Gambar 2.2 Pinout RHF0M301 Lora Gateway	4
Gambar 2.3 Raspberry PI 3 B+	6
Gambar 2.4 Pinout Raspberry Pi 3 B+.....	6
Gambar 2.5 Logo Python	8
Gambar 2.6 Contoh GUI Tkinter dari Beberapa Sistem Operasi.....	8
Gambar 3.1 Model Perancangan	9
Gambar 3.2 Skematik Gateway.....	11
Gambar 3.3 Desain GUI pada Tampilan Layar	12
Gambar 3.4 Rancangan Posisi Node Dalam Map	13
Gambar 3.5 Flowchart Sistem Gateway.....	14
Gambar 3.6 Flowchart Pengolahan Data Node Menjadi GUI	15
Gambar 4.1 Dimensi awal aplikasi.....	17
Gambar 4.2 Tampilan aplikasi.....	18
Gambar 4.3 Tombol node saat ditekan.....	18
Gambar 4.4 Menu combo box "Node 1".....	20
Gambar 4.5 Menu combo box "Node 2"	20
Gambar 4.6 Menu combo box "Node 3"	21
Gambar 4.7 Menu combo box "View All".....	21
Gambar 4.8 Posisi End Device Node 1.....	22
Gambar 4.9 Posisi End Device Node 2.....	23
Gambar 4.10 Posisi End Device Node 3.....	23
Gambar 4.11 Semua posisi node pada map.....	24

DAFTAR TABEL

	Halaman
Tabel 2.1 Keterangan pinout RHMFOm301 LoRa Gateway	5
Tabel 2.2 Spesifikasi RHF0M301	5
Tabel 3.1 Tabel Pin Connector Raspberry Pi 3 B+ dan RHF0m301	11
Tabel 4.1 Struktur data yang diterima LoRa Gateway.....	26
Tabel 4.2 Standar Packet Loss Menurut TIPHON.....	28
Tabel 4.3 Hasil Pengujian Keakuratan Terima Data Node 1.....	28
Tabel 4.4 Hasil Pengujian Keakuratan Terima Data Node 2.....	28
Tabel 4.5 Hasil Pengujian Keakuratan Terima Data Node 3.....	29
Tabel 4.6 Hasil Pengambilan data angket pengalaman pengguna aplikasi.....	31
Tabel L2.1 Data Node 1 Terkirim.....	44
Tabel L2.2 Data Node 1 diterima LoRa Gateway.....	50
Tabel L3.1 Data Node 2 Terkirim.....	57
Tabel L3.2 Data Node 2 diterima LoRa Gateway.....	63
Tabel L4.1 Data Node 3 Terkirim.....	71
Tabel L4.2 Data Node 3 diterima Gateway.....	77



UNIVERSITAS
Dianamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Program GUI	37
Lampiran 2 Tabel Uji Keakuratan Data Node 1	44
Lampiran 3 Tabel Uji Keakuratan Data Node 2	57
Lampiran 4 Tabel Uji Keakuratan Data Node 3	71
Lampiran 5 Hasil Turnitin.....	85

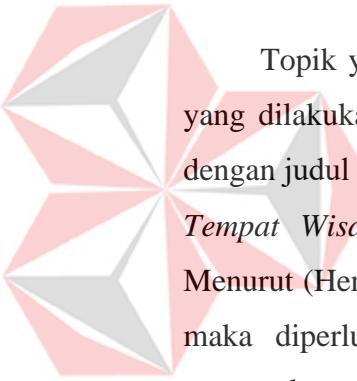


BAB I

PENDAHULUAN

1.1 Latar Belakang

Internet of Thing (IoT) adalah sebuah konsep dimana suatu objek yang memiliki kemampuan untuk mentransfer data melalui jaringan tanpa memerlukan interaksi manusia ke manusia atau manusia ke komputer. IoT telah berkembang dari konvergensi teknologi nirkabel, *micro-electromechanical systems* (MEMS), dan Internet. IoT sangat erat hubungannya dengan komunikasi mesin dengan mesin (M2M) tanpa campur tangan manusia ataupun komputer yang lebih dikenal dengan istilah cerdas (smart). Badan Siber dan Sandi Negara (BSSN) memprediksi bahwa ditahun 2021 pengguna *Internet of Things* (IoT) akan lebih banyak dibandingkan dengan pengguna Smartphone yang terkoneksi (Mathilda, 2020).



Topik yang diambil pada tugas akhir ini merupakan bagian dari penelitian yang dilakukan oleh Program studi S1 Teknik Komputer Universitas Dinamika dengan judul “*Sistem Komunikasi Jarak Jauh untuk Pelacakan Posisi Pengunjung Tempat Wisata dalam Upaya Peningkatan Keamanan Pariwisata Nasional*”. Menurut (Henaulu, et al., 2016) Untuk menunjang kelancaran industri pariwisata, maka diperlukan sinergitas yang baik antara pemerintah, stakeholder, dan masyarakat setempat (lokasi wisata) dalam memenuhi 5 (lima) pilar pariwisata, 5 (lima) pilar tersebut diantaranya: daya tarik (*attraction*), kemudahan akses (*accessibility*), infrastruktur (*amenity*), dan faktor pendukung lainnya (*ancillary*), serta diperlukan keterlibatan masyarakat dan pihak keamanan dalam menjaga stabilitas guna pengembangan pariwisata berkelanjutan (*sustainable tourism development*).

Terkait mengenai sistem IoT yang akan dibangun perangkat LPWAN yang digunakan adalah LoRa RF95 dan LoRa Gateway RHF0M301. Kedua alat tersebut masing – masing berperan sebagai *end device* (node) dan *gateway* dengan skenario *Multipoint to Point*. Node pada tugas akhir kali ini merupakan alat yang dapat berpindah dari satu tempat ke tempat lain sehingga dibutuhkannya pemantauan data node untuk mengetahui posisi pengunjung demi meningkatkan sektor keamanan

pada industri pariwisata. Pemantauan data node harus disajikan dengan tampilan yang akurat dan mudah dimengerti oleh pemonitor tempat wisata, tujuannya agar penanganan saat terjadi kecelakaan terhadap pengunjung dapat dilakukan dengan cepat.

GUI (*Graphic User Interface*) adalah suatu sistem yang membuat para pengguna atau user mempunyai berinteraksi dengan suatu perangkat komputer yang digunakan oleh user. GUI sendiri dapat dikendalikan menggunakan beberapa macam alat input, seperti mouse, keyboard, touchscreen, dan lain sebagainya (pelayananpublik, 2020). Maka dengan membangun sistem IoT menggunakan GUI sangatlah efektif dalam menangani permasalahan pemantauan data node. Data yang telah diterima *LoRa Gateway* dari node harus divisualkan sehingga user yang terlibat pada sistem IoT kali ini dapat memantau data node dengan mudah.



1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, diperoleh rumusan permasalahan sebagai berikut :

1. Bagaimana merancang tampilan data alamat atau identitas dan lokasi node pada LoRa Gateway menggunakan sistem GUI?
2. Bagaimana menguji keakuratan data lokasi node pada LoRa Gateway yang menggunakan sistem GUI?

1.3 Batasan Masalah

Dalam tugas akhir ini, ruang lingkup penelitian dibatasi dengan:

1. Node menggunakan sistem *Microcontroller* terintegrasi *Microcontroller*, baterai, dan modul GPS.
2. Gateway menggunakan sistem terintegrasi dengan *Microcontroller* yang bersifat *single board computer* dan berperan sebagai pencatat data (*data logger*).
3. Visibilitas data node hanya bisa dilihat oleh satu user yaitu pemonitor.

4. Sistem IoT yang dibangun pada tugas akhir ini hanya komunikasi antar *gateway* dan tiga node.

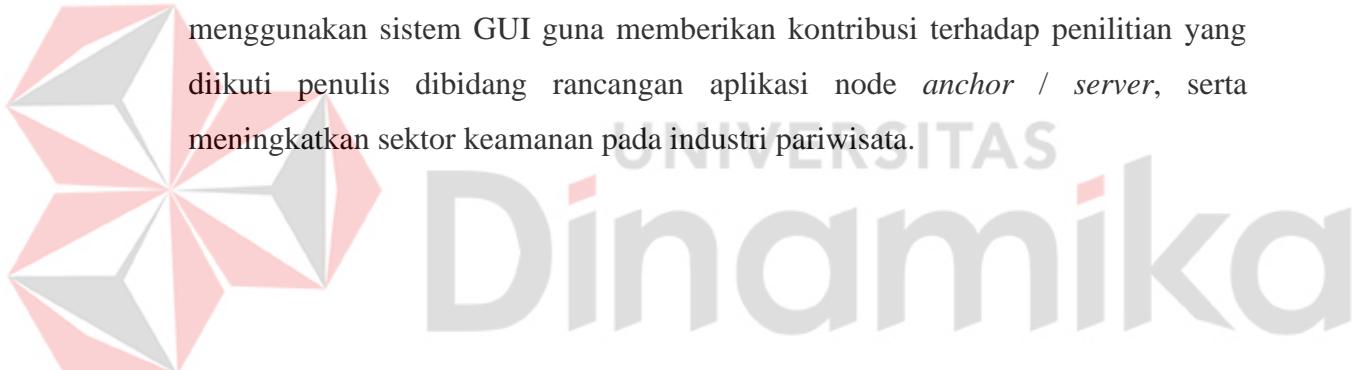
1.4 Tujuan

Tujuan dari tugas akhir ini adalah sebagai berikut :

1. Merancang tampilan data alamat dan lokasi node pada LoRa Gateway menggunakan sistem GUI.
2. Menguji keakuratan data lokasi node pada LoRa Gateway yang menggunakan sistem GUI.

1.5 Manfaat

Adapun manfaat dari tugas akhir ini adalah merancang aplikasi yang memuat tampilan data node dan menguji keakuratan data node pada LoRa Gateway yang menggunakan sistem GUI guna memberikan kontribusi terhadap penilitian yang diikuti penulis dibidang rancangan aplikasi node *anchor / server*, serta meningkatkan sektor keamanan pada industri pariwisata.



BAB II

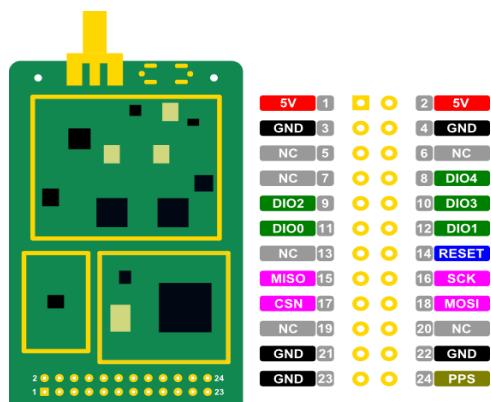
LANDASAN TEORI

2.1 RHF0M301 LoRa Gateway



Gambar 2.1 RHF0M301 LoRa Gateway
(Sumber : IoTiger.net, 2017)

RHF0M301 merupakan *LoRa Gateway* berperforma tinggi dengan ukuran yang kecil berdimensi 6,3 cm x 4,0cm. Modul LoRa Gateway ini tertanam SX1301 yang merupakan chip *gateway*, serta memiliki 24 pin yang berfungsi sebagai penghubung terhadap *Microcontroller*.



Gambar 2.2 Pinout RHF0M301 Lora Gateway
(Sumber : IoTiger.net, 2017)

Tabel 2.1 Keterangan pinout RHF0M301 LoRa Gateway

Nama	Tipe	Deskripsi
VCC 5V	POWER (VCC)	+5V Input
VCC 5V	POWER (VCC)	+5V Input
GND	POWER (GND)	Ground
GND	POWER (GND)	Ground
NC	-	No connection
NC	-	No connection
NC	-	No connection
SX1301_GPIO4	I/O	GPIO4 from SX1301
SX1301_GPIO3	I/O	GPIO2 from SX1302
SX1301_GPIO2	I/O	GPIO3 from SX1303
SX1301_GPIO0	I/O	GPIO0 from SX1304
SX1301_GPIO1	I/O	GPIO1 from SX1305
NC	-	No connection
RESET	I	Reset signal input to reset SX1301
MISO	O	MISO of SPI
SCK	I	SCK of SPI
CSN	I	CSN of SPI
MOSI	I	MOSI of SPI
NC	-	No connection
NC	-	No connection
GND	POWER (GND)	Ground
GND	POWER (GND)	Ground
GND	POWER (GND)	Ground
GPS_PPS	INPUT	PPS from GPS to module

(Sumber : risinghf.com, 2017)

Adapun modul ini diharapkan dapat menunjang sistem jaringan nirkabel, *AMR*, dan aplikasi IoT lainnya. Modul ini dapat bekerja pada frekuensi 434MHz, 470 - 510MHz, 780MHz, 868MHz dan 915MHz.

Tabel 2.2 Spesifikasi RHF0M301

Keterangan	Spesifikasi
Catu daya	< 6V
Rentang Temperatur	-40°C sampai +85°C
Dimensi	40 x 63 mm
RF input power	< -13dBm
Antarmuka	SPI

(Sumber : risinghf.com, 2017)

2.2 Raspberry PI 3 B+



Gambar 2.3 Raspberry PI 3 B+
(Sumber : pngkit.com)

Raspberry Pi 3 Model B+ merupakan sebuah komputer mini yang bisa digunakan untuk perangkat komputer maupun proyek-proyek menarik lainnya. Raspberry Pi meningkatkan kinerja Model B+ ini dengan menggunakan chipset baru yaitu Broadcom BCM2837B0 Cortex A53 64-bit berkecepatan 1,4GHz. Perangkat ini juga dilengkapi 40 pin GPIO (*General Purpose Input Output*) berfungsi layaknya sebuah *Microcontroller* yang memiliki PWM, I/O dan *interface* seperti UART, SPI dan I2C (Sorayakit, 2016).



Gambar 2.4 Pinout Raspberry Pi 3 B+
(Sumber : initialboard.com, 2021)

2.3 GPS (*Global Positioning System*)

Global Positioning System (GPS) adalah sistem navigasi berbasis satelit yang terdiri dari setidaknya 24 satelit. GPS berfungsi dalam segala kondisi cuaca, di mana pun di dunia, 24 jam sehari, tanpa biaya berlangganan atau biaya penyiapan. Departemen Pertahanan AS (USDOD) awalnya menempatkan satelit ke orbit untuk penggunaan militer, tetapi mereka dibuat tersedia untuk digunakan sipil pada 1980-an. Satelit GPS mengelilingi Bumi dua kali sehari dalam orbit yang tepat. Setiap satelit mengirimkan sinyal unik dan parameter orbital yang memungkinkan perangkat GPS untuk memecahkan kode dan menghitung lokasi tepat dari satelit. Penerima GPS menggunakan informasi dan trilaterasi ini untuk menghitung lokasi pasti pengguna. Pada dasarnya, penerima GPS mengukur jarak ke masing-masing satelit dengan jumlah waktu yang diperlukan untuk menerima sinyal yang dikirimkan. Dengan pengukuran jarak dari beberapa satelit lagi, penerima dapat menentukan posisi pengguna dan menampilkannya secara elektronik

2.4 GUI (*Graphical User Interface*)

GUI merupakan salah satu teknologi yang dibuat dan di teliti sejak lama, akan tetapi publik baru menikmatinya di awal tahun 1980an. Penggunaan dari teknologi ini merupakan sebuah teknologi yang digunakan untuk menggantikan sistem yang lama yang sering disebut dengan CLI. CLI atau *Command Line Interface* merupakan sebuah teknologi yang dibangun untuk memperbolehkan user memberikan perintah pada komputer dalam bentuk bahasa pemrograman. Dengan menggunakan CLI ini, inputan yang bisa diterima oleh komputer berasal dari keyboard saja.

GUI merupakan salah satu jenis user interface yang digunakan untuk melakukan komunikasi antara manusia dengan perangkat seperti laptop, komputer, ponsel dan tablet. Hal ini menjadikan komponen GUI selalu berhubungan dengan representasi visual dari sebuah sistem operasi ataupun software.

2.5 Python

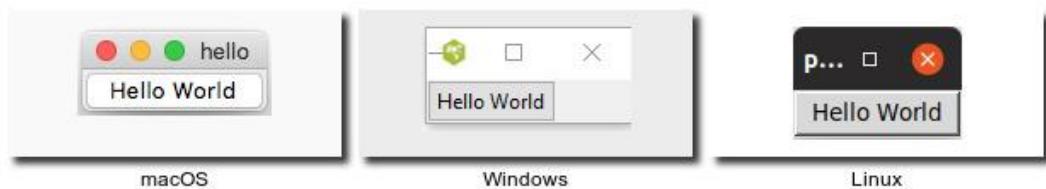


Gambar 2.5 Logo Python

Python adalah bahasa pemrograman yang populer. Bahasa pemrograman ini dibuat oleh Guido van Rossum dan dikenalkan sejak tahun 1991. Python termasuk bahasa pemrograman yang mudah untuk dipelajari. Sampai saat ini bahasa pemrograman Python hampir dipakai di segala bidang seperti game, sistem berbasis web, dan bahkan dapat membuat mesin pencari sendiri. Jadi secara umum, bahasa pemrograman ini dipakai dalam pengembangan website, pengembangan *software*, matematika, dan *system scripting*.

2.5.1 Tkinter

Tkinter adalah library standar Python untuk membuat aplikasi GUI atau desktop. Tkinter merupakan bentuk OOP dari Tcl/Tk. Tcl (*Tool Command Language*) adalah sebuah bahasa pemrograman dan TK adalah library yang digunakan oleh Tcl untuk membuat aplikasi GUI.

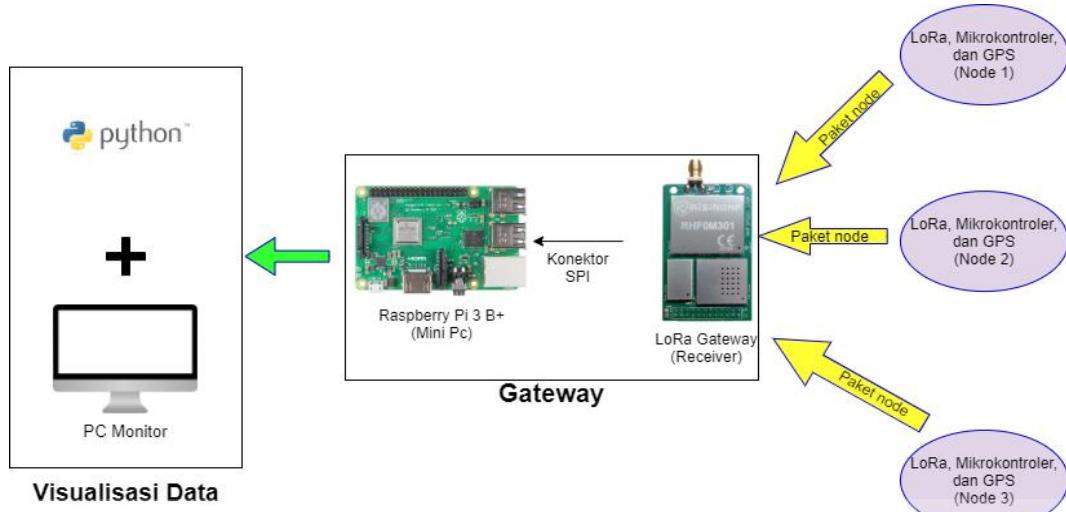


Gambar 2.6 Contoh GUI Tkinter dari Beberapa Sistem Operasi
(Sumber : petanikode.com, 2019)

BAB III

METODOLOGI PENELITIAN

3.1 Model Perancangan



Gambar 3.1 Model Perancangan

Skenario pada Gambar 3.1 menggunakan tiga buah perangkat **node** dan satu buah perangkat *LoRa gateway* tersambung dengan Raspberry Pi 3 B+ yang terintegrasi dengan PC Monitor. Setiap perangkat tersebut memiliki peran dan tugasnya masing-masing dan penunjang bagi perangkat lainnya.

3.1.1 Node

Node merupakan perangkat LoRa yang dirancang untuk mengirim data GPS ke *gateway* melalui medium modulasi LoRa. GPS dan modul LoRa yang secara urut terintegrasi dengan ADC dan *interface SPI Microcontroller* yang digunakan pada setiap perangkat node. Dalam skenario perancangan, terdapat tiga buah node yang mengirim data GPS, setiap node secara statis disematkan alamat untuk membedakan antara node satu dengan lainnya.

3.1.2 Receiver

Receiver atau bisa juga disebut *Gateway*, merupakan perangkat LoRa yang dirancang untuk menerima data GPS dari setiap node. Perangkat ini tidak menggunakan baterai sebagai sumber catu daya. Agar dapat berkomunikasi dengan

node, *gateway* menggunakan modul RHF0M301 yang terintegrasi dengan *interface SPI* pada *Microcontroller*.

Dalam skenario perancangan jumlah node lebih dari satu, maka *LoRa Gateway* diharapkan mampu menerima data-data yang dikirimkan dari banyak node dalam waktu yang sama dan mampu membedakannya. Pada tugas akhir ini penulis akan melakukan eksplorasi cara *LoRa Gateway* menerima data dari banyak node.

3.1.3 Mini-PC

Mini-PC yang digunakan pada model perancangan Gambar 3.1 merupakan perangkat bernama Raspberry Pi 3 B+. *Microcontroller* disini berperan sebagai pengelola paket data yang diterima oleh *Receiver* untuk ditampilkan pada *PC Monitor*. Selain mengelola data *mini-pc* juga berperan sebagai eksekutor program yang digunakan untuk mengoperasikan *LoRa Gateway*, sehingga data node dapat dibedakan antar satu sama lain.

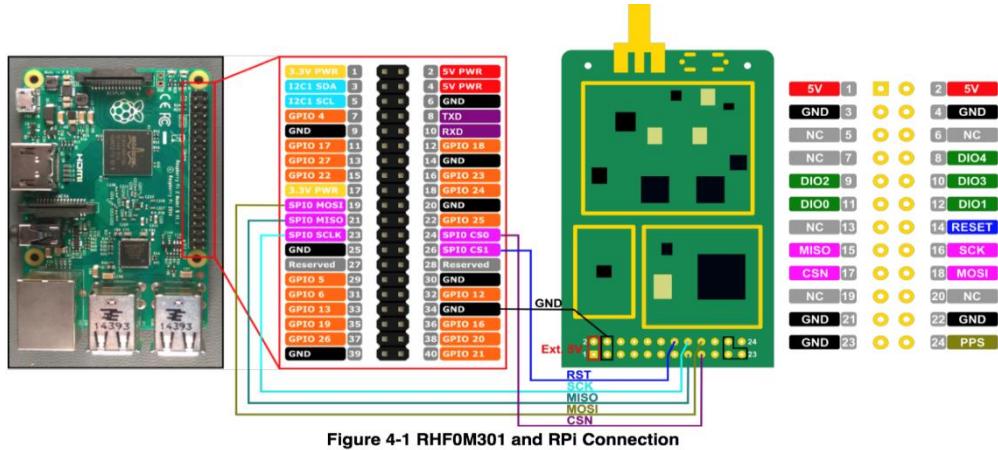
3.1.4 Visualisasi Data

Visualisasi Data pada gambar 3.1 merupakan hasil pengolahan yang telah dilakukan oleh sistem *Gateway*. *Platform* untuk pengembangan tampilan atau dapat disebut juga *GUI (Graphic User Interface)* menggunakan *software Python*. Sedangkan perangkat yang digunakan untuk menampilkan data olahan dari *Gateway* adalah *PC Monitor*. *PC Monitor* merupakan perangkat yang berperan sebagai pemantau posisi node. *Software* atau perangkat lunak yang digunakan diharapkan dapat menunjang *interface* bagi penggunanya dalam memantau pergerakan node dengan didukung fitur-fitur yang tersedia didalam Python.

Pembuatan *GUI* dilakukan pada *mini-pc* menggunakan library Python, yaitu Tkinter. Tampilan atau *GUI* yang akan dibuat diharapkan dapat mencakup semua informasi node, seperti alamat node dan isi dari modulasi data node. Setelah informasi diatas didapatkan, maka user dengan mudah mngetahui lokasi node.

3.2 Skematik Perangkat Keras

3.2.1 Skematik Gateway



Gambar 3.2 Skematik Gateway
(Sumber : thethingsnetwork.org/forum, 2016)

Seperti yang dijelaskan sebelumnya pada model perancangan, perangkat *gateway* dilengkapi oleh modul RHF0M301 yang tersambung dengan Raspberry Pi 3 B+ agar dapat berkomunikasi dengan node dan menegelola data menjadi tampilan. Skematik *gateway* dapat dilihat pada Gambar 3.2.

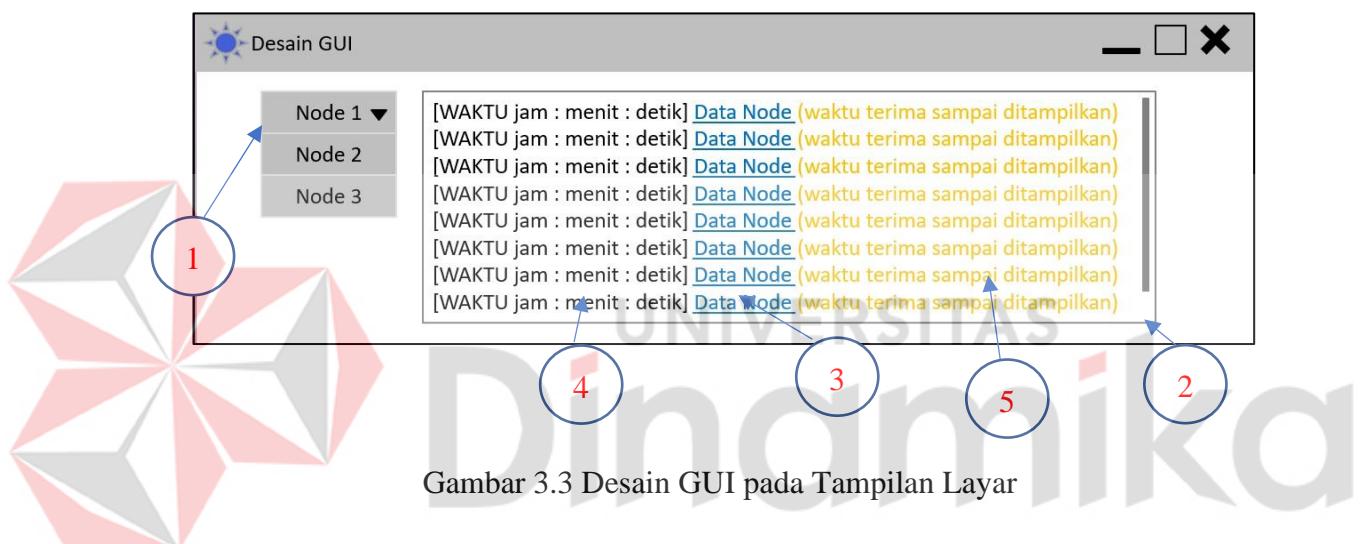
Tabel 3.1 Tabel Pin Connector Raspberry Pi 3 B+ dan RHF0m301

Raspberry Pi 3 B+	RHF0M301
SPI0 MOSI	MOSI
SPI0 MISO	MISO
SPI0 SCLK	SCK
SPI0 CS0	CSN
SPI0 CS1	RST

3.3 Perancangan GUI

Dalam perancangan aplikasi untuk membaca data node diperlukan komponen -komponen GUI yang mendukung dalam menvisualkan data node yang diterima oleh LoRa Gateway. Melalui komponen – komponen GUI maka data node yang merupakan data GPS akan divisualisasikan menjadi posisi node didalam map. Map atau peta yang digunakan merupakan lokasi wisata yang ingin dipantau. Berikut uraian mengenai lingkup perancangan aplikasi yang akan dibuat pada tugas akhir ini.

3.3.1 Desain GUI

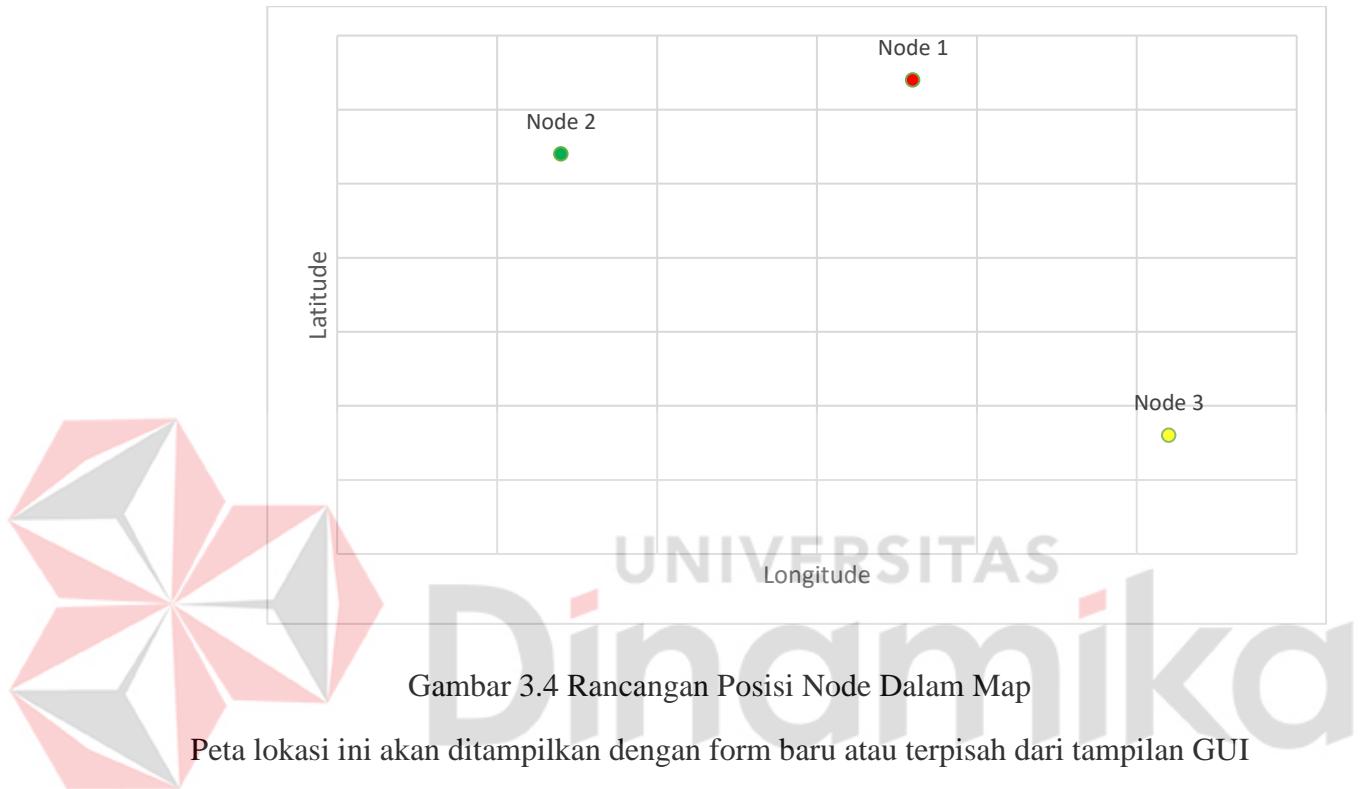


Adapun bagian-bagian GUI yang merupakan sarana interaksi manusia terhadap mesin daripada *mini-pc* dapat dilihat pada Gambar 3.3. Berikut bagian - bagiannya :

1. *Combo box / box ganda*, Apabila di-click akan menampilkan opsi Node yang akan dipantau datanya. Dalam aplikasi ini node yang dapat dipantau adalah Node 1, Node 2, dan Node 3.
2. *Multi text box* menampilkan serta mencatat waktu terima data, data GPS
3. Data node dapat di-click, kemudian menampilkan posisi node yang berada didalam map atau lokasi tempat wisata yang dipantau.
4. Waktu terima *gateway* terhadap data node.
5. Perhitungan waktu terima data node sampai dengan ditampilkan kedalam GUI.

3.3.2 Posisi Node Dalam Map

Adapun visualisasi node ke dalam peta lokasi dilakukan setelah pengguna aplikasi men-click data node. Peta lokasi menggunakan grafik dengan masing – masing koordinat *latitude* dan *longitude* peta memiliki nilai maksimal yang sama dengan *latitude* dan *longitude* dari peta lokasi sesungguhnya.

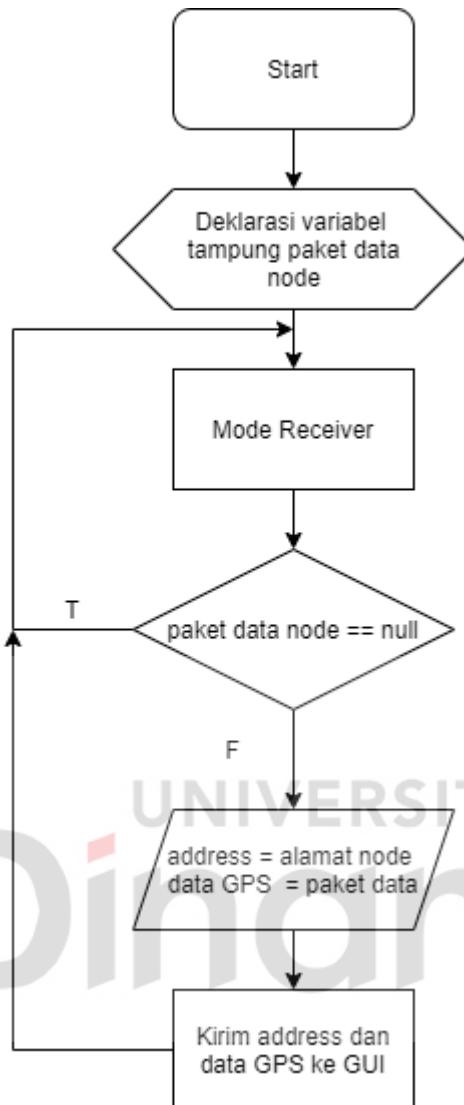


Peta lokasi ini akan ditampilkan dengan form baru atau terpisah dari tampilan GUI pada Gambar 3.3. Setelah posisi node pada peta lokasi yang dipantau diketahui, maka pengguna yang bertindak sebagai pengawas lokasi wisata diharapkan bisa menuju ke lokasi node dengan mudah bila terjadi bahaya terhadap pengunjung.

3.4 Perancangan Perangkat Lunak

Dalam perancangan perangkat lunak, terdapat tiga program yang mengacu pada aliran data dan alamat dari node menuju *gateway*, membedakan alamat atau data node satu dengan yang lainnya, dan pengelolaan data alamat dan lokasi node yang diterima LoRa Gateway untuk ditampilkan ke dalam GUI.

3.4.1 Algoritma Sistem Gateway



Gambar 3.5 Flowchart Sistem Gateway

Seperti yang dijelaskan sebelumnya, peran *gateway* sangatlah penting terhadap tersampaikannya paket data node menuju *Microcontroller* sehingga dapat dikelola dan ditampilkan ke dalam GUI. Algoritma sistem *Gateway* pada Gambar 3.4 dimulai dari deklarasi variabel untuk menampung paket data node kemudian mengaktifkan *mode receiver* pada *gateway*, tujuannya agar peran *gateway* hanya sebagai penerima.

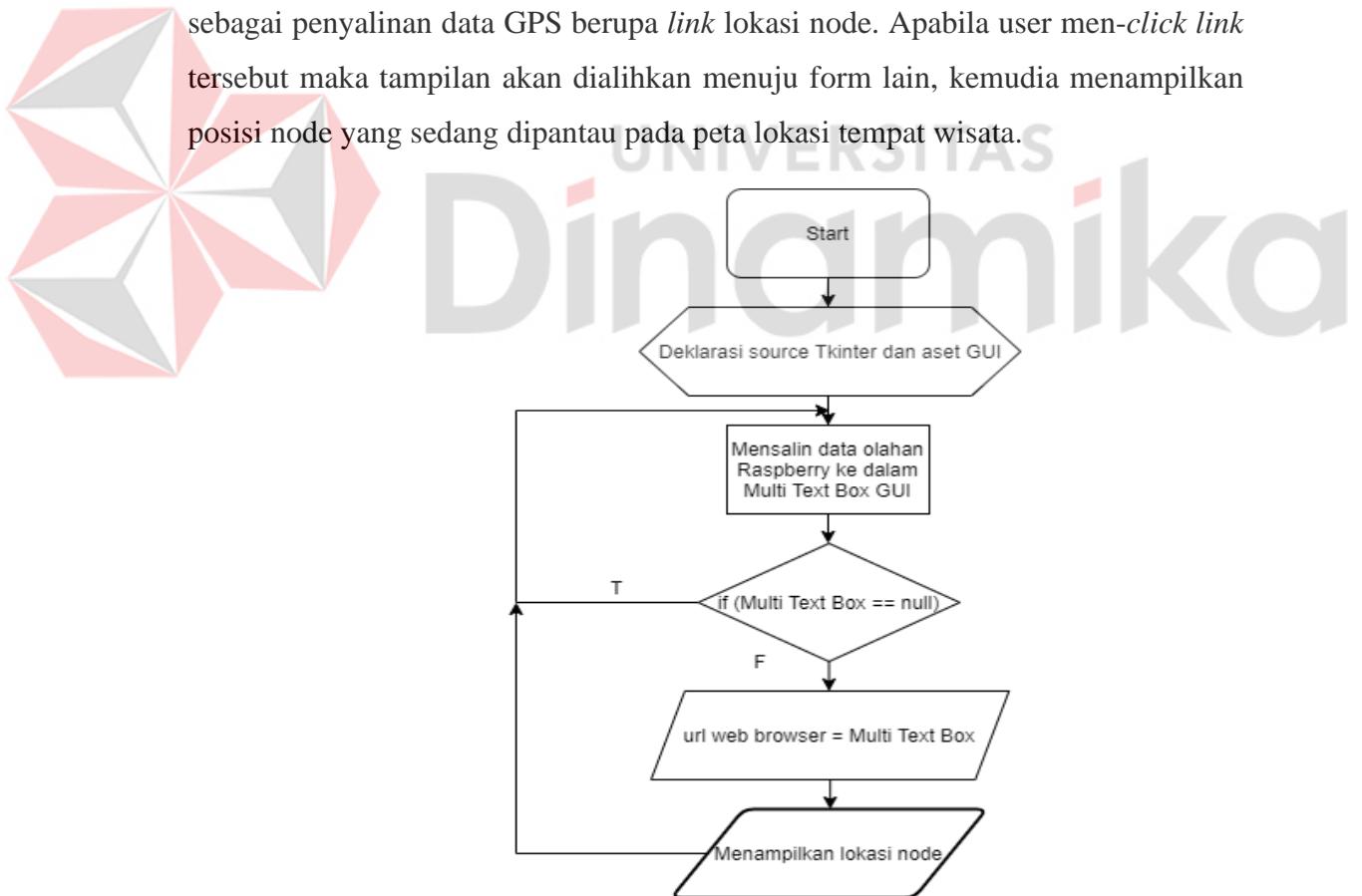
Setelah menerima paket data node maka dilakukan pengecekan terhadap isi dari paket data node. Jika sama dengan *null* atau kosong maka sistem akan kembali

lagi ke mode receiver *gateway* dan jika tidak kosong maka paket data node akan dipisah menjadi dua data yaitu alamat dan data GPS. Kedua data ini nantinya akan dikelola untuk ditampilkan ke dalam GUI.

3.4.2 Pengolahan Data Node Menjadi GUI

Seperti yang telah disebutkan sebelumnya, *PC Monitor* hanya menampilkan data node pada GUI dan pembuatan GUI dilakukan pada *mini-pc*. Sebelum dapat melakukan hal tersebut *library* atau *resource* yang diperlukan harus disiapkan terlebih dahulu. Kemudian data node yang terdiri dari alamat atau identitas node dan GPS ditentukan tipe datanya. Hal ini dilakukan agar dapat di-*sinkronkan* antara data olahan terhadap GUI.

Data node yang telah di-*sinkronkan* terhadap GUI akan disalin kedalam *multi-text box* GUI. Program yang disematkan pada *multi-text box* GUI berfungsi sebagai penyalinan data GPS berupa *link* lokasi node. Apabila user men-*click* *link* tersebut maka tampilan akan dialihkan menuju form lain, kemudia menampilkan posisi node yang sedang dipantau pada peta lokasi tempat wisata.



Gambar 3.6 Flowchart Pengolahan Data Node Menjadi GUI

BAB IV

HASIL DAN PEMBAHASAN

Pada bab ini terdapat hasil berupa aplikasi pemantau data node berbasis GUI (*Graphic User Interface*) serta cara mengoperasikannya, pengujian keakuratan data yang diterima oleh LoRa Gateway, dan hasil angket pengalaman pengguna setelah mengoperasikan aplikasi yang dirancang oleh penulis.

4.1 Aplikasi Pemantau Data Node

Adapun aplikasi berbasis pemantau data node berbasis GUI dibuat menggunakan bahasa pemrograman Python dengan menggunakan *library* Tkinter sebagai pendukung pembuatannya. Struktur pemrograman dalam pembuatan aplikasi pemantau data node berbasis GUI dibagi menjadi dua bagian utama yaitu GUI dan data node *receiver*.

4.1.1 Program GUI Menggunakan Tkinter

Adapun pembuatan GUI menggunakan Tkinter memerlukan inisialisasi sebagai berikut :

A. Inisialisasi

```
from tkinter import *
from tkhtmlview import *
import time
import sys
import lorthon
import json
import threading
```

Baris program diatas merupakan pemanggilan *library* yang dibutuhkan untuk membuat aplikasi berbasis GUI ini.

B. Inisialisasi Komponen GUI

Komponen – komponen yang diperlukan untuk membuat aplikasi pemantau data node berbasis GUI adalah dimensi awal, tombol, *combo box*, *check button*, dan *label*.

```
root = Tk()
root.title('PKPT UNDIKA')
root.geometry("900x400")
```

Program diatas merupakan dimensi awal untuk meletakkan komponen – komponen lain.



Gambar 4.1 Dimensi awal aplikasi



```
# Monitor DATA
Monitor = tk.scrolledtext.ScrolledText(root, width=50,
height=20)
monitor.config(state = DISABLED)
monitor.pack(side = RIGHT, fill = BOTH, expand = True)

# combo box (node)
clicked = StringVar()
clicked.set(options[0])
drop = OptionMenu(root, clicked, *options)
drop.pack(padx = 10)
#---drop.grid(row = 0,column = 0,columnspan = 1)

# check Button
var = IntVar()
autoscroll = tk.Checkbutton(root, text= "Auto Scroll", variable
= var)
autoscroll.select()
autoscroll.pack(padx = 10)
```

Setelah inisialisasi dimensi awal maka baris program diatas merupakan penempatan komponen aplikasi *combo box*, *check button*, dan *textbox* seperti pada Gambar 4.2 dibawah.



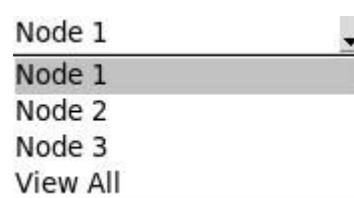
Gambar 4.2 Tampilan aplikasi

C. Function pada GUI

Terdapat beberapa *function* untuk menjalankan fungsi dari komponen – komponen GUI.

```
drop = ttk.Combobox(root,
                     values=[  
                     "Node 1",  
                     "Node 2",  
                     "Node 3",  
                     "View All",])
```

Baris program diatas merupakan array yang dipanggil oleh variabel “drop” untuk membuat opsi pilihan Node 1, Node 2, Node 3, dan View All saat tombol node ditekan.



Gambar 4.3 Tombol node saat ditekan

C.1 Function display_data()

```

def display_data():
    ch = '1234567890-.,'
    Fixedgps = ''
    A = len(app.y)
    for I in range(A):
        if app.y[i] not in ch:
            print(app.y[i] in ch)

        else:
            print(app.y[i] in ch)
            fixedgps = fixedgps + app.y[i]

    start_time = time.time()
    monitor.tag_config("tag",foreground = "blue")
    monitor.tag_config("tag1",foreground = "green")
    monitor.tag_config("tag2",foreground = "orange")
    monitor.tag_config("tag3",foreground = "red")
    monitor.config(state = NORMAL)
    hour = time.strftime("%H")
    minute = time.strftime("%M")
    second = time.strftime("%S")
    if clicked.get() == "Node 1" and len(fixedgps[2:]) <= 20 :
        if fixedgps[:2] == '01':
            monitor.insert(tk.END, "[")
            monitor.insert(tk.END, hour, "tag1")
            monitor.insert(tk.END, ":")

            monitor.insert(tk.END, minute, "tag1")
            monitor.insert(tk.END, ":")

            monitor.insert(tk.END, second, "tag1")
            monitor.insert(tk.END, "] ")
            monitor.insert(tk.END, "Node 1 : ")
            monitor.tag_bind("tag", "<Button-1>", callback)
            monitor.tag_bind("tag", "<Enter>", show_cursor)
            monitor.tag_bind("tag", "<Leave>", normal_cursor)
            monitor.insert(tk.END, "www.google.com/maps/place/","tag")
            monitor.insert(tk.END, fixedgps[2:], "tag")
            monitor.insert(tk.END, "(executed : %.2f second)"%(time.time() -start_time),"tag2")
            monitor.insert(tk.END, "\n")

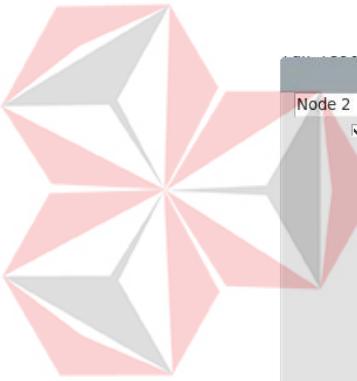
        .
        .
        .

        monitor.config(state = DISABLED)
        if var.get() == 1:
            monitor.see(tk.END)

```

Setelah pengguna aplikasi menekan tombol node maka *text box* akan menerima data sesuai dengan kondisi nilai dari variabel “drop”. Jika nilai dari variabel “drop” adalah “Node 1” dan dua bit awal Data *end device* node 1 adalah “01” maka *text box* hanya menerima data dari *end device* node 1,

begitu juga dengan kondisi “Node 2” dan “Node 3” masing -masing dengan dua bit di awal data adalah “02” dan “03”.

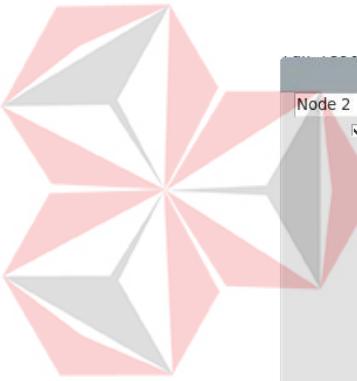


```

PKPT UNDIKA
Node 1
Auto Scroll
[20:26:56] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:26:58] Node 1 : -7.450673,112.728694 (executed : 0.04 second)
[20:26:59] Node 1 : -7.450673,112.728694 (executed : 0.11 second)
[20:27:00] Node 1 : -7.450673,112.728694 (executed : 0.11 second)
[20:27:01] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:02] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:03] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:04] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:05] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:06] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:07] Node 1 : -7.450673,112.728694 (executed : 0.01 second)
[20:27:08] Node 1 : -7.450673,112.728694 (executed : 0.01 second)
[20:27:09] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:10] Node 1 : -7.450673,112.728694 (executed : 0.08 second)
[20:27:11] Node 1 : -7.450673,112.728694 (executed : 0.01 second)
[20:27:12] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:13] Node 1 : -7.450673,112.728694 (executed : 0.01 second)
[20:27:14] Node 1 : -7.450673,112.728694 (executed : 0.05 second)
[20:27:15] Node 1 : -7.450673,112.728694 (executed : 0.06 second)
[20:27:17] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:18] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:19] Node 1 : -7.450673,112.728694 (executed : 0.04 second)
[20:27:20] Node 1 : -7.450673,112.728694 (executed : 0.00 second)
[20:27:21] Node 1 : -7.450673,112.728694 (executed : 0.00 second)

```

Gambar 4.4 Menu combo box "Node 1"



```

PKPT UNDIKA
Node 2
Auto Scroll
[20:28:54] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:28:55] Node 2 : -7.450701,112.728765 (executed : 0.06 second)
[20:28:56] Node 2 : -7.450701,112.728765 (executed : 0.04 second)
[20:28:57] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:28:58] Node 2 : -7.450701,112.728765 (executed : 0.04 second)
[20:28:59] Node 2 : -7.450701,112.728765 (executed : 0.04 second)
[20:29:00] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:01] Node 2 : -7.450701,112.728765 (executed : 0.05 second)
[20:29:02] Node 2 : -7.450701,112.728765 (executed : 0.15 second)
[20:29:03] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:04] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:05] Node 2 : -7.450701,112.728765 (executed : 0.03 second)
[20:29:06] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:07] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:08] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:09] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:10] Node 2 : -7.450701,112.728765 (executed : 0.05 second)
[20:29:12] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:13] Node 2 : -7.450701,112.728765 (executed : 0.00 second)
[20:29:14] Node 2 : -7.450701,112.728765 (executed : 0.00 second)
[20:29:15] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:16] Node 2 : -7.450701,112.728765 (executed : 0.01 second)
[20:29:17] Node 2 : -7.450701,112.728765 (executed : 0.04 second)
[20:29:18] Node 2 : -7.450701,112.728765 (executed : 0.14 second)

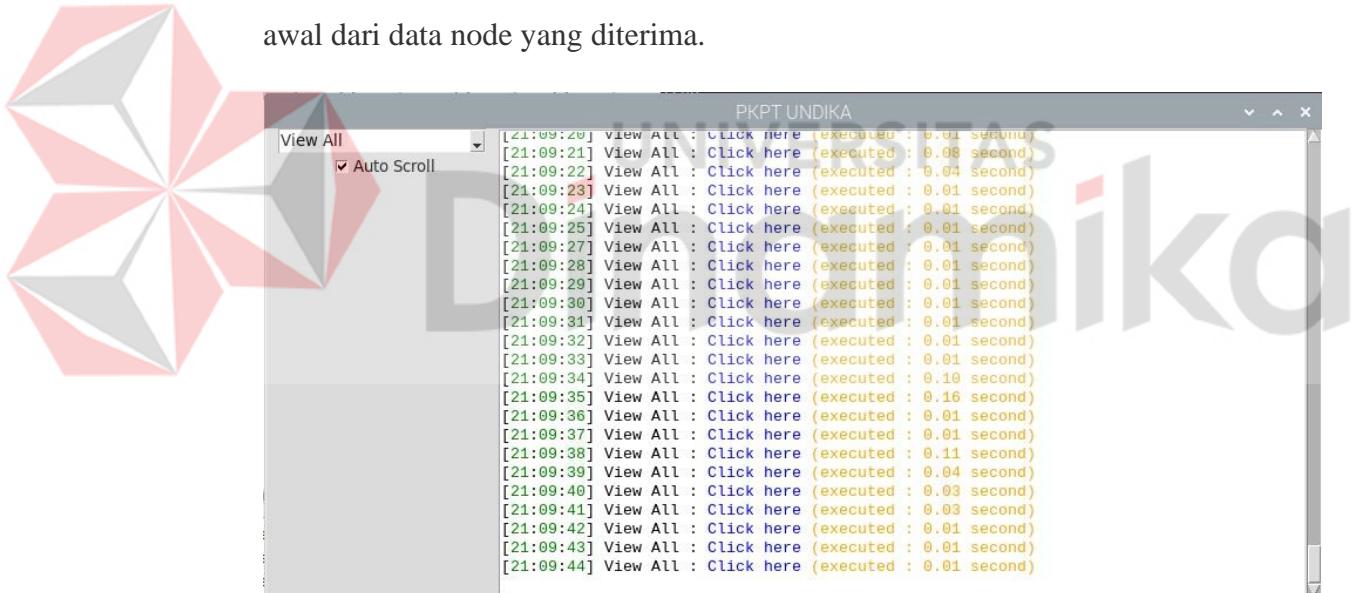
```

Gambar 4.5 Menu combo box "Node 2"



Gambar 4.6 Menu combo box "Node 3"

Untuk pemantauan menggunakan menu “View All” tidak dibutuhkan 2 bit awal dari data node yang diterima.



Gambar 4.7 Menu combo box "View All"

C.2Function callback(event)

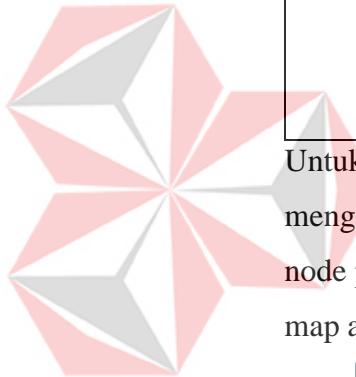
```
Def callback(event):
    index = event.widget.index("@%s,%s" % (event.x, event.y))
    tag_indices = list(event.widget.tag_ranges('tag'))

    for start, end in zip(tag_indices[0::2], tag_indices[1::2]):
```

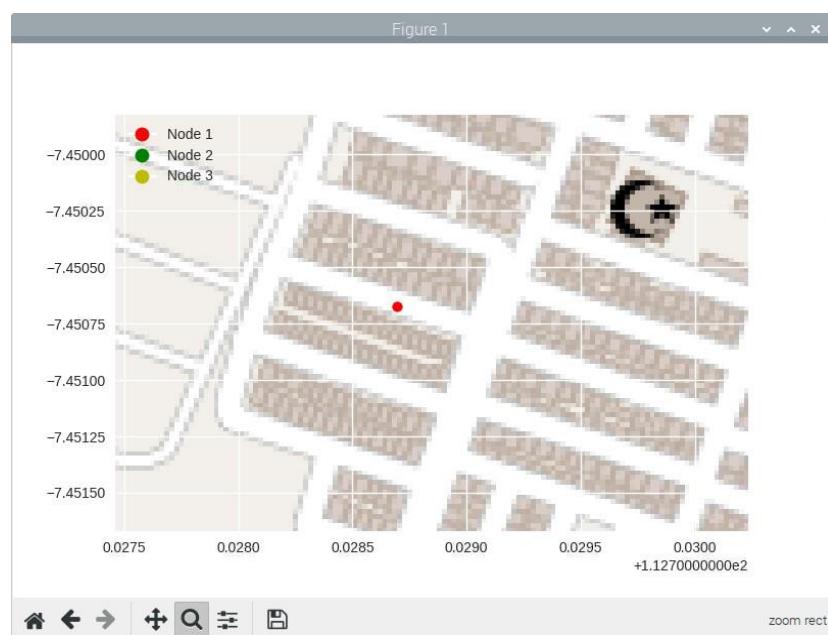
```

        if event.widget.compare(start, '<=', index) and
event.widget.compare(index, '<', end):
    # return string between tag start and end
    print(start , end , event.widget.get(start, end))
    if clicked.get() == "Node 1":
        .
        .
        .
    if clicked.get() == "Node 2":
        .
        .
        .
    if clicked.get() == "Node 3":
        .
        .
        .
    if clicked.get() == "View All":
        .
        .
        .

```



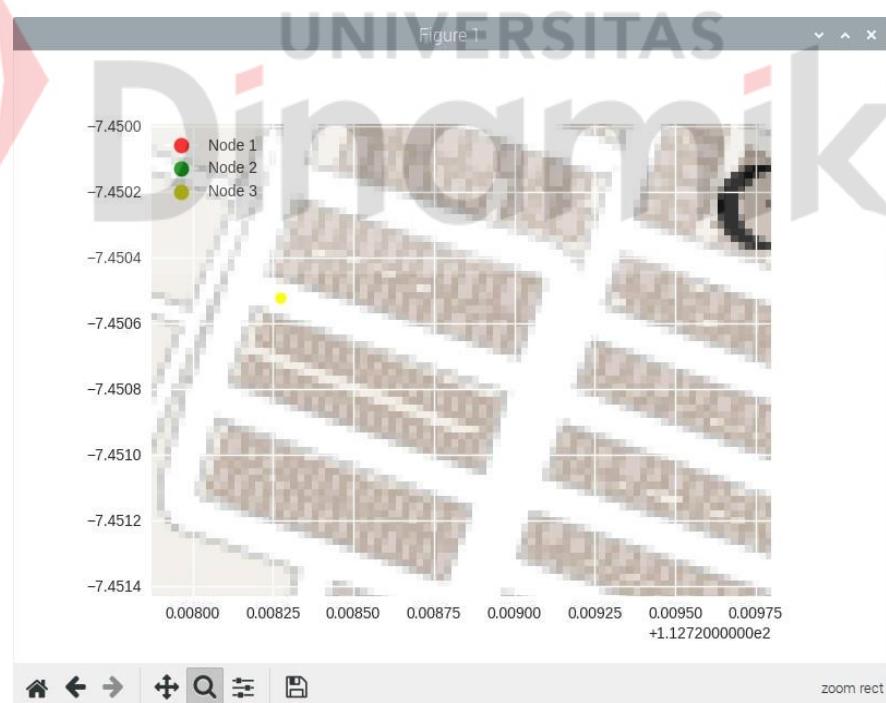
Untuk baris program diatas merupakan function yang bekerja untuk mengidentifikasi posisi data node yang di *click* pada *text box*. Setelah data node pada *multi textbox* di-click maka form baru berisikan posisi node pada map akan ditampilkan diluar tampilan GUI utama.



Gambar 4.8 Posisi End Device Node 1



Gambar 4.9 Posisi End Device Node 2



Gambar 4.10 Posisi End Device Node 3



Gambar 4.11 Semua posisi node pada map

Pada saat menu “View All” dipilih maka akan menampilkan semua posisi node didalam map seperti pada Gambar 4.11

4.1.2 LoRa Gateway sebagai receiver

Perangkat LoRa Gateway merupakan alat yang bisa melakukan komunikasi secara *half duplex* (satu arah) maupun *full duplex* (dua arah). Pada tugas akhir kali ini LoRa Gateway akan difungsikan sebagai *receiver* atau penerima. Data yang diterima oleh LoRa Gateway merupakan data GPS yang dikirimkan node *end device*.

Untuk menjalankan LoRa Gateway seri RHF0301 pada *mini-pc* dibutuhkan library khusus yaitu “lorthonpy”. Berikut program untuk LoRa Gateway sebagai receiver.

```
if (lorthon.py_LoRaInit("global_conf.json") == 0):
    while True:
        y=str()
        value = json.loads(lorthon.py_LoRaRx())
        for data in value['packets']:
```

```
y = data['payload']
print(bytes.fromhex(y).decode('utf-8','ignore'))
```

Program diatas merupakan cara untuk menjadikan LoRa Gateway sebagai *receiver* atau penerima. Pertama, LoRa Gateway melakukan pengecekan terhadap “global_config.json” dimana file ini merupakan tempat diterimanya data masuk. Jika file “global_config.json” belum menerima data node maka dilanjutkan dengan melakukan penyaringan data node yang terdiri dari beberapa variabel secara terus menerus. Data yang ditampilkan aplikasi berasal dari variabel “payload” Berikut contoh beberapa variabel dari file “global_config.json” ketika menerima data dari node.

```
{
  "packets": [
    {
      "frequency": 914800000,
      "if_chain": 1,
      "status": 17,
      "count_us": 74555419,
      "rf_chain": 0,
      "modulation": 16,
      "bandwidth": 3,
      "datarate": 2,
      "coderate": 1,
      "rss": -119,
      "snr": -12,
      "snr_min": -13.5,
      "snr_max": -9,
      "crc": 30822,
      "size": 45,
      "payload": "xff/xff/30312d372e343438383430392c3131322e3732343035x00/x00/x00/x00/x00/x00/x00/x00/x00....."
    }
  ],
  "number_of_pkts": 1
}
```

}

Data “payload” diatas merupakan bilangan *hexadesimal* yang apabila dikonversi kedalam ASCII (*American Standard Code for Information Interchange*) akan menjadi “ÿþ01-7.4488409,112.72405”. berikut keterangan mengenai struktur data yang diterima LoRa gateway.

Tabel 4.1 Struktur data yang diterima LoRa Gateway

Karakter / Data Node	Keterangan
ÿþ	Prosedur
01	Identitas data node
-7.4488409,112.72405	Data node

Jika data yang diterima oleh LoRa Gateway tidak memiliki prosedur seperti yang ditunjukkan pada Tabel 4.1, maka dapat dipastikan data tersebut bukan dari node.

4.1.3 Threading

Program GUI dan program data *node receiver* tidak bisa dijalankan secara sekuensial. Hal ini disebabkan keduanya masing-masing menjalankan tugasnya dengan struktur *looping*. Maka solusi yang diberikan adalah memasukkan program data *node receiver* ke dalam “class threading”. Hal ini dilakukan agar program GUI dan program data *node receiver* dapat berjalan secara *multitasking*. Berikut program “class threading” dari data *node receiver*.

```
class App(threading.Thread):
    def __init__(self):
        threading.Thread.__init__(self)
        self.start()

    def all_quit(self):
        app.quit()
```

```

def run (self):
    if (lorthon.py_LoRaInit("global_conf.json") == 0):
        while True:
            value = json.loads(lorthon.py_LoRaRx())
            x = ''
            for data in value['packets']:
                print (data['payload'])
                x =
            bytes.fromhex(data['payload']).decode('utf-8','ignore')
            self.y = x
            display_data()

```

Dalam “class App(threading.Thread)” terdapat tiga *function* yaitu “*_init_(self)*” yang merupakan prosedur atau inisiasi untuk melakukan “threading”. Parameter “*self*” yang digunakan pada setiap *function* bertujuan agar variabel didalamnya dapat diakses diluar “class App(threading.Thread)”.

Function “*all_quit(self)*” adalah protokol agar program data node *receiver* tidak mengganggu saat GUI ditutup. Apabila function ini tidak ada maka “class App()” tidak menghentikan kerjanya setelah GUI ditutup, melainkan secara bersamaan sehingga terjadi *crash* antar program.

Kemudian didalam function “*run (self)*” merupakan program data node *receiver* yang telah disebutkan sebelumnya dengan tambahan pemanggilan function “*display_data()*” agar data node yang diterima LoRa Gateway dieksekusi dan ditampilkan oleh program GUI.

4.2 Uji coba keakuratan terima data dan kecepatan waktu proses tampilan

Adapun pengujian keakuratan terima data yang dilakukan menggunakan tiga *end device* node secara bersamaan, serta perhitungan kecepatan waktu proses tampilan sebagai evaluasi terhadap kinerja Raspberry Pi 3 B+. Masing-masing node mengirimkan data GPS menuju LoRa Gateway.

4.2.1 Uji coba keakuratan terima data

Skema pengujian keakuratan data adalah membandingkan antara data node yang dikirimkan dengan data yang diterima LoRa Gateway. Setelah melakukan pengujian maka akan didapatkan informasi keakuratan terima data LoRa Gateway berupa *packet loss* (paket hilang) dalam bentuk persentase. Adapun standar *packet loss* menurut TIPHON adalah :

Tabel 4.2 Standar Packet Loss Menurut TIPHON

Kategori <i>Packet Loss</i>	<i>Packet Loss (%)</i>
<i>Poor</i>	>25%
<i>Medium</i>	12 – 24%
<i>Good</i>	3 – 14%
<i>Perfect</i>	0 – 2%

Sumber : (NN, 2002)

Setelah dilakukan uji coba pengiriman data menuju LoRa Gateway menggunakan tiga node maka didapati hasil pengujian sebagai berikut :

Tabel 4.3 Hasil Pengujian Keakuratan Terima Data Node 1

NODE 1 => Gateway	
Jarak	100 meter
Waktu	5 menit
Error	2
Loss	14
Data terkirim	265
Data diterima	251
Persentase loss	6,04%
Rata-rata waktu proses	0,045 detik

Tabel 4.4 Hasil Pengujian Keakuratan Terima Data Node 2

NODE 2 => Gateway	
jarak	100 meter
waktu	5 menit
error	0
loss	3
data terkirim	266
data diterima	263
persentase loss	1,13%

Rata-rata waktu proses 0,09 detik

Tabel 4.5 Hasil Pengujian Keakuratan Terima Data Node 3

NODE 3 => Gateway	
jarak	100 meter
waktu	5 menit
error	2
loss	3
data terkirim	268
data diterima	265
persentase loss	1,9%
Rata-rata waktu proses	0,04 detik

Melihat tabel hasil uji coba keakuratan terima data node diatas dapat diambil kesimpulan bahwa aplikasi memantau data node 1 tergolong dalam kriteria *good* atau baik dengan persentase nilai loss sebesar 6,04%, kemudian node 2 tergolong dalam kriteria *perfect* atau sangat baik dengan persentase nilai loss sebesar 1,13%, dan node 3 tergolong dalam kriteria *perfect* atau sangat baik juga dengan persentase nilai loss sebesar 1,9% menurut TIPHON.

4.2.2 Evaluasi kecepatan waktu proses tampilan

Adapun uji coba mengenai kecepatan aplikasi dalam menampilkan data node perlu dievaluasi dikarenakan *mini-pc* yang digunakan mempunyai memori kecil. Perhitungan dilakukan menggunakan timer yang berjalan ketika data node diterima, kemudian berhenti ketika data node sudah ditampilkan ke dalam GUI.

Melihat hasil rata-rata waktu eksekusi dari Tabel 6, Tabel 7, dan Tabel 8 berturut – turut adalah 0.045, 0.09, dan 0.04 detik maka dapat diambil kesimpulan bahwa waktu proses *mini-pc* dalam menampilkan data node sangat cepat.

4.3 Angket pengalaman pengguna setelah mengoperasikan aplikasi

Pengambilan data angket pengalaman pengguna dilakukan dengan menyebarkan 13 lembar angket kepada orang secara acak. Kriteria angket menggunakan terdiri dari 5 jenis yaitu (1)Sangat setuju, (2)Setuju, (3)Normal, (4)Tidak setuju, (5)Sangat tidak setuju. Setiap angket terdiri dari 18 pertanyaan dengan parameternya masing-masing yang terbagi menjadi 7 jenis yaitu :

1. *Text and Font Usage* : 2 pertanyaan
2. *Background & Foreground* : 3 pertanyaan
3. *Button* : 3 pertanyaan
4. *Menu Structure & Navigation* : 2 pertanyaan
5. *Learnability* : 2 pertanyaan
6. *Flexibility* : 3 pertanyaan
7. *Robustness* : 3 pertanyaan

Adapun parameter angket pengalaman pengguna aplikasi yang digunakan adalah *Text and Font Usage*, *Background & Foreground*, *Button*, and *Menu Structure & Navigation* merupakan faktor *design*.

Sedangkan parameter angket pengalaman pengguna aplikasi pada faktor usability adalah *Learnability*, *Flexibility*, and *Robustness*. Berikut hasil pengambilan data angket pengalaman pengguna aplikasi :



Tabel 4.6 Hasil Pengambilan data angket pengalaman pengguna aplikasi

No	Pertanyaan	Jumlah Jawaban Responden					Percentase					Parameter	Jawaban Terbanyak
		Sangat tidak setuju	Tidak setuju	Normal	Setuju	Sangat Setuju	Sangat tidak setuju	Tidak Setuju	Normal	Setuju	Sangat setuju		
1	Pertanyaan 1	0	1	3	4	5	0,00%	7,69%	23,08%	30,77%	38,46%		Sangat setuju
2	Pertanyaan 2	1	1	1	5	5	7,69%	7,69%	7,69%	38,46%	38,46%		Sangat setuju & Setuju
3	Pertanyaan 3	0	0	4	6	3	0,00%	0,00%	30,77%	46,15%	23,08%		Setuju
4	Pertanyaan 4	1	0	1	5	6	7,69%	0,00%	7,69%	38,46%	46,15%	Background And Foreground	Sangat setuju
5	Pertanyaan 5	1	1	1	5	5	7,69%	7,69%	7,69%	38,46%	38,46%		Sangat setuju & Setuju
6	Pertanyaan 6	0	0	3	3	7	0,00%	0,00%	23,08%	23,08%	53,85%		Sangat setuju
7	Pertanyaan 7	0	2	0	6	5	0,00%	15,38%	0,00%	46,15%	38,46%	Button	Setuju
8	Pertanyaan 8	0	2	1	2	8	0,00%	15,38%	7,69%	15,38%	61,54%		Sangat setuju
9	Pertanyaan 9	0	0	4	3	6	0,00%	0,00%	30,77%	23,08%	46,15%	Menu Sturcture And Navigation	Sangat setuju
10	Pertanyaan 10	0	0	1	2	10	0,00%	0,00%	7,69%	15,38%	76,92%		Sangat setuju
11	Pertanyaan 11	0	2	3	5	3	0,00%	15,38%	23,08%	38,46%	23,08%		Setuju
12	Pertanyaan 12	0	0	4	4	5	0,00%	0,00%	30,77%	30,77%	38,46%	Learnability And User Friendly	Sangat setuju
13	Pertanyaan 13	0	0	3	6	4	0,00%	0,00%	23,08%	46,15%	30,77%		Setuju
14	Pertanyaan 14	0	0	3	3	7	0,00%	0,00%	23,08%	23,08%	53,85%	Flexibility	Sangat setuju
15	Pertanyaan 15	0	1	2	7	3	0,00%	7,69%	15,38%	53,85%	23,08%		Setuju
16	Pertanyaan 16	0	1	2	7	3	0,00%	7,69%	15,38%	53,85%	23,08%		Setuju
17	Pertanyaan 17	1	1	2	4	5	7,69%	7,69%	15,38%	30,77%	38,46%	Robustness	Sangat setuju
18	Pertanyaan 18	1	1	0	7	4	7,69%	7,69%	0,00%	53,85%	30,77%		Setuju
												Percentase Jawaban terbanyak =	Sangat setuju = 61%
													Setuju = 50%

Berdasarkan Tabel 4.6 dapat menunjukkan jawaban terbanyak dari 13 orang responden secara acak dari 18 pertanyaan yang diajukan. Pilihan jawaban yang banyak dipilih oleh responden adalah (1) Sangat Setuju dan (2) Setuju. Untuk mencari nilai persentase dari jawaban (1) Sangat Setuju menggunakan rumus berikut:

$$\begin{aligned}
 \text{Sangat Setuju} (\%) &= \frac{\text{Jawaban terbanyak (1) Sangat Setuju}}{\text{Jumlah semua pertanyaan}} \times 100\% \\
 &= \frac{11}{18} \times 100\% \\
 &= 61\%
 \end{aligned}$$

Sedangkan untuk mencari nilai persentase dari jawaban (2) Setuju menggunakan rumus berikut :

$$\begin{aligned}
 \text{Setuju} (\%) &= \frac{\text{Jawaban terbanyak (2) Setuju}}{\text{Jumlah semua pertanyaan}} \times 100\% \\
 &= \frac{9}{18} \times 100\% \\
 &= 50\%
 \end{aligned}$$

Sehingga dapat diambil kesimpulan bahwa faktor *design* dan *usability* mendapatkan jawaban terbanyak (1) Sangat Setuju sebanyak 11 pertanyaan dengan persentase 61% dan (2) Setuju sebanyak 9 pertanyaan dengan persentase 50% dari 18 pertanyaan, sehingga aplikasi pemantau data node memenuhi syarat HCI (*Human Computer Interaction*).

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Adapun rancang bangun aplikasi berbasis GUI (*Graphic User Interface*) untuk pembacaan data pada LoRa Gateway menghasilkan beberapa kesimpulan, yaitu :

1. Aplikasi berbasis GUI untuk pembacaan data pada LoRa Gateway dapat memantau data dan membedakan ketiga data node sekaligus.
2. Aplikasi dapat menvisualkan data node berupa posisi lokasi pengunjung didalam peta berbasis grafik *scatter-plot*.
3. Uji coba keakuratan terima data tiga node terhadap LoRa, node 1 tergolong dalam kriteria *good* atau baik dengan persentase nilai loss sebesar 6,04%, kemudian node 2 tergolong dalam kriteria *perfect* atau sangat baik dengan persentase nilai loss sebesar 1,13%, dan node 3 tergolong dalam kriteria *perfect* atau sangat baik juga dengan persentase nilai loss sebesar 1,9%. Sehingga data node 1, node 2, dan node 3 secara keseluruhan dapat diterima dengan baik oleh aplikasi berbasis GUI pemantau data node.
4. Pengambilan data angket pengalaman pengguna aplikasi menghasilkan kesimpulan bahwa aplikasi berbasis GUI (*Graphic User Interface*) untuk pembacaan data pada LoRa Gateway memenuhi syarat HCI (*Human Computer Interaction*) dengan jawaban terbanyak “sangat setuju” sebanyak 11 pertanyaan dengan persentase 61% dan “setuju” sebanyak 9 pertanyaan dengan persentase 50% dari 18 pertanyaan.

5.2 Saran

1. Untuk selanjutnya aplikasi pemantau data node dapat dikembangkan ke *smartphone* sehingga pemantauan data node bersifat lebih *portable*.
2. Aplikasi ini dapat memantau lebih dari 3 data node.
3. Apilksi ini dapat diimplementasikan pada bidang keamanan publik.
4. Penambahan fitur dalam aplikasi, seperti penambahan tombol save untuk menyimpan data node yang telah dipantau.

DAFTAR PUSTAKA

- Ajie, 2019. *Bikin IOT Client Paling Mudah dengan ESP-01 dan Module Relay*. [Online]
- Available at: <http://saptaji.com/2019/05/01/bikin-iot-client-paling-mudah-dengan-esp8266-esp01-dan-relay-module/>
- Android, 2021. *Mengenal Android Studio*. [Online]
- Available at: <https://developer.android.com>
- arrow, 2020. *Semtech SX1301IMLRT*. [Online]
- Available at: <https://www.arrow.com>
- Arthur, 2017. PEMODELAN SISTEM PELACAKAN LOT PARKIR KOSONG BERBASIS SENSOR ULTRASONIC DAN INTERNET OF THINGS (IOT) PADA LAHAN PARKIR DILUAR JALAN. *Seminar Nasional Sains dan Teknologi 2017*, pp. 6-8.
- Feriakviansyah, 2018. *PENGERTIAN DARI INTERMEDIARY DEVICES,END DEVICES,NETWORK MEDIA*. [Online]
- Available at: <http://feriakviansyah.blogspot.com/>
- Haidar, 2019. Pengembangan Sistem Perantara Pengiriman Data Menggunakan Modul Komunikasi LoRa dan Protokol MQTT Pada Wireless Sensor Network. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 1657-1658.
- Henaulu, A. K., Syairudin, B. & Gunarta, I. K., 2016. *STRATEGI PENGEMBANGAN INDUSTRI PARIWISATA BERBASIS*. Surabaya, Institut Teknologi Sepuluh Nopember, pp. A-31-2.
- HopeRF, 2019. *RF Modules*. [Online]
- Available at: <https://www.hoperf.com/>
- id, P. p., 2020. *Apa itu GUI, Sejarah, Fungsi Hingga Contohnya*. [Online]
- Available at: <https://pelayananpublik.id/2020/04/20/apa-itu-gui-sejarah-fungsi-hingga-contohnya/>

IoTiger, 2020. *RHF0M301 – LORA GATEWAY AND CONCENTRATOR MODULE.* [Online]

Available at: <http://iotiger.net/>

Irwandi, S., 2018. *Raspberry Pi 3 Model B+, Dilengkapi dengan Chipset Quad Core dan WiFi Dual Band.* [Online]

Available at: <https://www.yangcanggih.com/>

Jeremiah, 2020. *Pengembangan Multi Komunikasi Perangkat M2M (Machine To Machine) Berbasis LoRa Pada Sistem IoT (Internet Of Things).* Surabaya: Universitas Dinamika.

Muhardian, A., 2019. *12 Modul Python untuk Membuat Aplikasi GUI (Desktop).*

[Online]

Available at: <https://www.petanikode.com/python-gui/>

Network, T. T., 2021. *Has anyone tried the RisingHF gateway boards?.* [Online]

Available at: <https://www.thethingsnetwork.org>

R, A., 2019. Implementasi Sistem Gateway Discovery pada Wireless Sensor Network (WSN) Berbasis Modul Komunikasi LoRa. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, pp. 2139-2142.

RisingHF, 2017. *RHF0M301 Datasheet.* NanShan district: risinghf.com.

Semtech, 2021. *Products Wireless RF.* [Online]

Available at: <https://www.semtech.com>

Yanziah, A., Soim, S. & Rose, M. M., 2020. Analisis Jarak Jangkauan Lora Dengan Parameter Rssi Dan Packet Loss Pada Area Urban. *Jurnal Teknologi Technoscientia*, p. vol.13 No.1.

Yida, 2020. *LoRa and LoRaWAN for Arduino and Raspberry Pi.* [Online]

Available at: <https://www.seeedstudio.com/>