

# Rancang Bangun Gelang Locator Pasien Terkonfirmasi Covid-19 dengan Metode Heversine sebagai Pengidentifikasi Pelanggaran Zona Aman Isolasi Mandiri Menggunakan Komunikasi Lora (Long Range)

Ahmad Afan Maulana<sup>1)</sup>, Riza Alfita<sup>2)</sup>, dan Achmad Fiqhi Ibadillah<sup>3)</sup>

<sup>1,2,3)</sup>Jurusan Teknik Elektro, Fakultas Teknik, Universitas Trunojoyo Madura

<sup>1,2,3)</sup>Jl. Raya Telang, Perumahan Telang Indah, kamal, Kabupaten Bangkalan, Jawa Timur

e-mail: [maulana07@live.com](mailto:maulana07@live.com)<sup>1)</sup>, [riza.alfitatrunojoyo@gmail.com](mailto:riza.alfitatrunojoyo@gmail.com)<sup>2)</sup>, [fiqhi.ibadillah@trunojoyo.ac.id](mailto:fiqhi.ibadillah@trunojoyo.ac.id)<sup>3)</sup>

## ABSTRACT

Positive cases of COVID-19 continue to grow rapidly. This prompted the government to make new rules in the form of various types of health protocols. One of them is the implementation of the self-isolation protocol for confirmed COVID-19 patients. Self-isolation is a rule that requires confirmed patients to stay at home until the specified time limit to break the chain of COVID-19 development. However, it is suggested and as a rule that there are still many ODP and PDP who still violate the rules of self-isolation from the government. The ODP and PDP still roam freely without thinking about the impacts and risks to their surroundings. With the departure of confirmed patients, the risk of developing COVID-19 becomes greater. In this study, a confirmed patient locator bracelet was made that can map the position of the patient based on latitude and longitude. The latitude and longitude data from the bracelet is sent to the gateway by LoRa communication. LoRa communication is also used to minimize the dimensions of the bracelet. In addition to mapping the location of the patients, by applying the Heversine method this system is able to identify when a safe zone violation occurs in the form of a change in the color of the marker. In the future, it is hoped that this system will be able to handle the problem of developing COVID-19 caused by confirmed patients who still violate the self-isolation protocol.

**Keywords:** API, COVID 19, Heversine, LoraWAN, Internet of Things

## ABSTRAK

Kasus positif COVID-19 terus saja berkembang pesat. Hal ini mendorong pemerintah untuk membuat aturan baru berupa protokol-protokol kesehatan yang berbagai macam jenisnya. Salah satunya ialah penerapan protokol isolasi mandiri yang diperuntukkan bagi para pasien terkonfirmasi COVID-19. Isolasi mandiri ialah aturan yang mewajibkan para pasien terkonfirmasi untuk tetap di rumah sampai batas waktu yang ditentukan untuk memutus rantai perkembangan COVID-19. Namun dihimbau dan menjadi aturan masih banyak para ODP dan PDP yang tetap saja melanggar aturan isolasi mandiri dari pemerintah. Para ODP dan PDP tetap saja berkeliaran bebas tanpa memikirkan dampak dan resiko terhadap sekitarnya. Dengan berkeliarnya para pasien terkonfirmasi maka resiko perkembangan COVID-19 menjadi besar. Pada penelitian ini dibuat gelang locator pasien terkonfirmasi yang dapat memetakan posisi para pasien berdasarkan *latitude* dan *longitude*. Data *latitude* dan *longitude* dari gelang dikirimkan ke *gateway* dengan komunikasi LoRa. Komunikasi LoRa dipakai juga untuk meminimalisir dimensi dari gelang. Selain memetakan lokasi para pasien, dengan menerapkan metode Heversine system ini mampu mengidentifikasi bila terjadi pelanggaran zona aman berupa perubahan warna marker. Kedepannya diharapkan system ini mampu mengani masalah perkembangan COVID-19 yang disebabkan oleh para pasien terkonfirmasi yang masih banyak melanggar protokol isolasi mandiri.

**Kata kunci:** API, COVID 19, Heversine, LoraWAN, Internet of Things.

## I. PENDAHULUAN

COVID-19 adalah wabah virus yang menjadi pandemik di seluruh dunia saat ini termasuk Indonesia. Berdasarkan statistik perkembangan covid per tanggal 11 Januari 2021 laju perkembangan pasien terkonfirmasi di Indonesia mencapai 828,026 pasien, dengan cluster penyebaran terbanyak terjadi di Provinsi Jakarta dan Jawa Timur. Bila dilihat dari statistik keseluruhan perkembangan COVID-19 di Indonesia yang dimulai dari awal Maret hingga September, perkembangan COVID-19 di Indonesia meningkat secara signifikan yang mana puncaknya terjadi pada tanggal 9 Januari 2021 yaitu sebanyak 10.617 pasien terkonfirmasi per harinya. Akibat dari hal ini pemerintah menerapkan Pembatasan Sosial Berskala

Besar atau yang familiar disebut PSBB. Hal ini sebagai upaya pemerintah untuk memutus mata rantai perkembangan COVID-19 di Indonesia. Dimulai awal Maret hingga awal Juni 2020 dilaksanakan PSBB di seluruh Indonesia. Namun tetap saja perkembangan COVID-19 di Indonesia masih saja terjadi peningkatan secara signifikan. Ini dikarenakan masih banyak masyarakat yang melanggar peraturan PSSBB dan masih banyak pasien terkonfirmasi yang berkeliaran keluar rumah.

Karena dalam penyebarannya COVID-19 tergolong mudah yaitu hanya melalui *droplet* atau percikan yang keluar dari hidung atau mulut dan menempel di benda seperti tangga, meja, dan sarana umum lainnya. Maka semakin banyak orang berkerumun atau berkumpul dalam suatu tempat maka semakin besar pula resiko

COVID-19 tersebar. Terutama bila di antara orang tersebut terdapat pasien terkonfirmasi COVID-19.

Sebenarnya permasalahan pertama dari pandemi ini ialah para pasien terkonfirmasi yang masih saja ada yang tidak mematuhi aturan dari pemerintah. Menyikapi ini pemerintah memberlakukan isolasi bagi para pasien terkonfirmasi di rumah sakit sebagai solusi agar pasien bisa dipantau dan ditertibkan. Namun masalah lain muncul akibat kebijakan tersebut, banyak rumah sakit mengalami overload ruang isolasi. Hal ini berimbas pada sarana dan prasarana public yang dialih fungsikan menjadi rumah sakit darurat untuk menampung pasien terkonfirmasi. Sebut saja wisma atlet yang sejak awal sudah dijadikan rumah sakit darurat oleh pemerintah dan masih banyak lagi sarana dan prasarana public yang beralih fungsi menjadi rumah sakit darurat.

Melihat masalah ini saya membuat penelitian yang berjudul Rancang Bangun Gelang Locator Pasien Terkonfirmasi COVID-19 Dengan Metode Heversine sebagai Pengidentifikasi Pelanggaran Zona Aman Isolasi Mandiri Menggunakan Komunikasi LoRa (Long Range). Cara kerja dari system ini sendiri ialah dengan mengirimkan koordinat pasien dengan menggunakan alat berupa gelang. Koordinat tersebut digunakan sebagai parameter untuk selanjutnya dikalkulasikan dengan koordinat zona aman yang telah ditentukan di awal. Selanjutnya dilakukan pengidentifikasi apakah pasien tersebut melanggar protokol isolasi mandiri atau tidak. Diharapkan dengan adanya penelitian ini, perkembangan COVID-19 di Indonesia bisa ditekan seminimal mungkin.

## II. TEORI

### A. COVID 19

COVID-19 adalah virus pertama ditemukan di Wuhan, China pada tahun 2019. Organisasi Kesehatan Dunia secara resmi menamai penyakit yang disebabkan oleh novel coronavirus sebagai penyakit coronavirus 2019 (COVID-19). Gejala awal yang dirasakan oleh penderita ialah seperti gejala flu seperti biasa sehingga banyak dari pemimpin Negara beranggapan bahwa virus ini tidak berbeda jauh dengan virus biasa. Namun gejala serius ditemukan dalam kurun waktu 14 hari terjangkit, gejala yang banyak muncul ialah demam, batuk sampai dengan sesak nafas bahkan ada beberapa dari penderita yang meninggal dunia.

### B. LoRa (Long Range)

LoRa merupakan salah satu protocol komunikasi nirkabel jarak jauh dan berdaya rendah. LoRa menjadi solusi baru untuk pengaplikasian Internet of Things yang membutuhkan infrastruktur minimal dengan hasil yang maksimal. LoRa memiliki dukungan langsung dari beberapa perusahaan teknologi diantaranya ialah IBM, Semtech, Actility, dll yang merupakan Aliansi dari LoRa Alliance. LoRa beroperasi di pita frekuensi 923,

868 atau 915, 920, 923 MHz, tergantung kebijakan setiap wilayah. Di Indonesia sendiri kominfo menetapkan frekuensi untuk penggunaan LoRa di Indonesia ialah di rentan frekuensi 915 – 923 MHz [1].

### C. LoRaWAN

LoRaWAN merupakan mekanisme kontrol akses menengah, yang memungkinkan banyak perangkat untuk berkomunikasi dengan gateway menggunakan modulasi LoRa. Secara umum, LoRaWAN adalah protokol yang digunakan untuk mengirimkan data dari beberapa node atau sensor menuju *network server* menggunakan protokol LoRa. Dari perspektif pengembangan aplikasinya jaringan LoRaWAN sangat cocok untuk digunakan dalam pengaplikasian IoT (Internet of Things) [2].

### D. GPS (Global Position System)

GPS adalah system navigasi yang digunakan untuk mengetahui detail suatu tempat atau informasi dengan cepat, tepat dan akurat berdasarkan data yang diterima dari satelit. GPS biasanya diaplikasikan dalam beberapa system, namun yang paling populer yaitu *system tracking* dimana gps akan mengirimkan data berupa *latitude* dan *longitu* untuk mengetahui posisi [7].

### E. API (Application Programming Interface)

API adalah seperangkat antarmuka pemrograman aplikasi yang dapat digunakan untuk mengembangkan aplikasi dalam satu platform ataupun lintas platform. API dapat berupa kumpulan *method* dan *class* dari suatu pustaka yang dinamakan Pustaka API. Secara garis besar API terbagi menjadi 3 kategori yaitu. [8] : Ownership WEB API, Communication Level API, WEB Service API.

### F. Metode Heversine

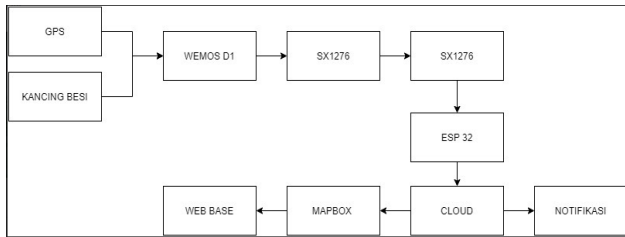
Metode Haversine adalah metode yang digunakan untuk melakukan kalkulasi jarak antara dua titik kordinat dalam sebuah area.

## III. METODE

Perancangan system ini menggunakan 2 tahap perancangan, yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*).

### A. Hardware

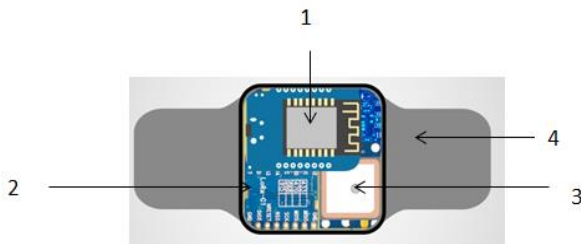
Dalam pembuatannya terdapat beberapa komponen yang digunakan sebagai penunjang pembuatan system, diantaranya yaitu: ESP32, esp 8266, gps neo blox 6, strap, sx1276, antenna 923Mhz, tiang bambu, box. Semua komponen disusun sesuai tahapan sehingga system bisa bekerja dengan baik.



Gambar 1. Diagram koneksi hardware

Pada diagram diatas terdapat 1 iput yaitu gps, gps mengirimkan *latitude* dan *longitude* menggunakan sx1276 sebagai LoRa transceiver ke ESP32 yang diterima dengan sx1276 sebagai LoRa receiver. Selanjutnya *latitude* dan *longitude* dikirim menuju *cloud* yang selanjutnya di parsing untuk selanjutnya ditampilkan di WEB menggunakan API Mapbox.

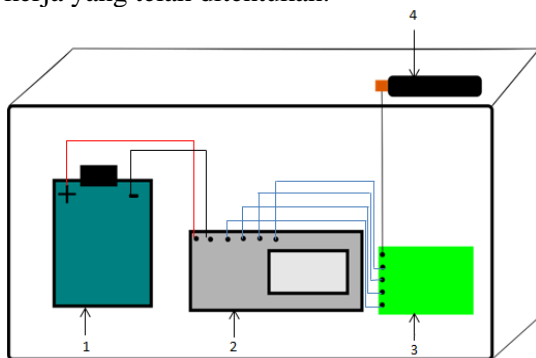
1. Node



Gambar 2. Desain Node

Berikut adalah penjelasan dan fungsi dari nomor di desain alat di atas

1. ESP8266 berfungsi sebagai mikrokontroler dan node untuk mentransmisikan data. Pemilihan ESP8266 dikarenakan esp8266 memiliki regulator 3,3v untuk mensupply sx1276 agar tidak terjadi over voltage serta karena dimensi ESP8266 memiliki dimensi yang memungkinkan diposisikan di atas dua komponen lainnya.
2. SX1276 berfungsi sebagai antenna yang digunakan sebagai transmisi data GPS ke gateway.
3. GPS Neo Blox-6 berfungsi sebagai pemberi koordinat *latitude* dan *longitude*.
4. Strap karet berfungsi sebagai gelang agar alat nyaman dipakai koordinat lubang pcb yang telah ditentukan. Raspberry juga terhubung dengan limit switch yang berfungsi untuk memberi batasan pada motor stepper agar tidak bekerja melewati bidang kerja yang telah ditentukan.

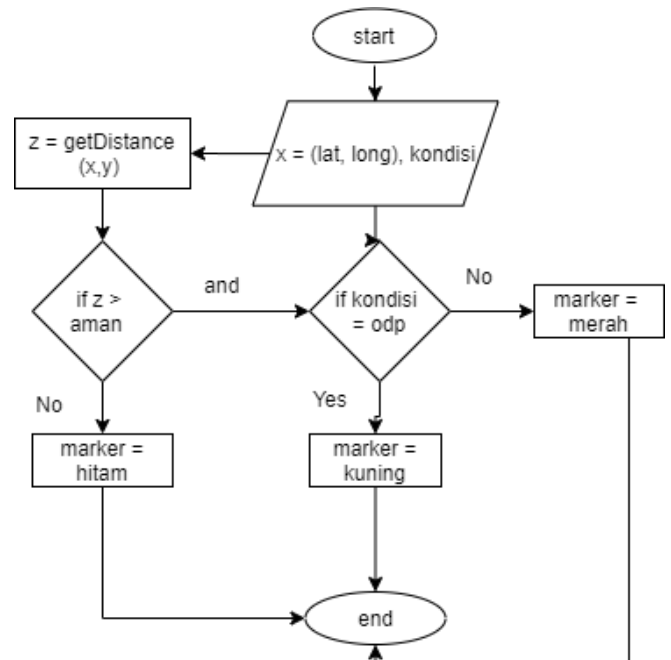


Gambar 3. Desain Gateway

Berikut merupakan penjelasan dari masing-masing komponen di atas

1. Baterai berfungsi sebagai supply ESP32.
2. ESP32 digunakan untuk mengirimkan data *latitude* dan *longitude* yang diterima dari beberapa node menuju server atau *cloud*.
3. Sx1276 digunakan sebagai antenna *receiver* untuk menerima data yang di transmisikan oleh beberapa node.
4. Antenna 915MHz digunakan sebagai penguat gelombang agar *gateway* dapat menerima beberapa transmisi data sekaligus dengan lancar.

## B. Software



Gambar 4. Flowchart Software

Program dimulai dengan mengambil data dari *cloud* dan database. Untuk data dari *cloud* yang diambil ialah *latitude* dan *longitude* disini saya beri nama *y*. dan data yang diambil dari database yaitu *latitude*, *longitude* dan *kondisi*, *latitude* dan *longitude* dari database diberi nama *x*, *x* adalah koordinat radius aman pasien sedangkan *y* ialah koordinat *real-time* yang didapat dari gps. Selanjutnya *x* dan *y* dikalkulasi menggunakan metode heversine untuk diketahui selisih jarak antara *x* dan *y*. Jika jarak hasil kalkulasi melebihi jarak aman maka marker akan berubah menjadi hitam, namun jika jarak masih berada dalam jarak aman maka marker akan berwarna sesuai dengan kondisi pasien, untuk ODP berwarna kuning dan PDP berwarna merah.

## IV. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini ialah gelang yang mampu mengirimkan data *latitude* dan *longitude* untuk selanjutnya dijadikan parameter untuk

mengidentifikasi pelanggaran zona aman oleh para pasien terkonfirmasi covid 19. Komponen utama dari alat ini ialah node dan gateway. Adapun ada beberapa tahapan pengujian yang dilakukan untuk mencapai keberhasilan alat yang diinginkan.

### A. Transmit Data Node

Pada percobaan pertama dilakukan pengiriman data dari node. Data yang dikirimkan berupa *latitude*, *longitude* yang berasal dari modul GPS neo 6 serta data tegangan kancing yang dihubungkan dengan pin D1 Wemos. Node yang digunakan disini sebanyak 3 node dengan spesifikasi yang sama untuk melihat sampai sejauh mana jarak yang dapat di receive oleh gateway dari ketiga node tersebut. Skema pengujian terbagi menjadi 3, yaitu dengan 1 node, 2 node dan yang terakhir 3 node. Hal ini untuk menguji respon gateway terhadap jumlah node yang di transmisikan.

Tabel I. Data hasil pengukuran jarak dengan 1 transmisi dari node 1. Node

ID	Waktu	Latitude	longitude	Jarak
001	14.53.20	-7.048762	113.934385	0m
001	14.56.50	-7.0469967	113.9300747	800m
001	14.59.10	-7.043621	113..923277	1.5km
001	15.03.40	-7.0371341	113.9104658	3.1km
001	15.06.13	-7.033686	113.907752	3.9km
001	15.10.30	-7.0389739	113.9001919	4.4km
001	15.14.50	-7.032920	113.890290	5.7km
001	15.17.08	-7.028128	113.881482	6.9km
001	14.20.57	-7.017139	113.879313	8.1km
001	14.25.01	-7.005676	113.861654	10.3km

Tabel II. Data hasil pengukuran transmit data dengan 2 node

ID	Waktu	Latitude	Longitude	Jarak
001	14.53.20	-7.048762	113.934385	0m
002	14.56.50	-7.0469967	113.9300747	800m
001	14.59.10	-7.043621	113..923277	1.5km
002	15.03.40	-7.0371341	113.9104658	3.1km
002	15.06.13	-7.033686	113.907752	3.9km
002	15.10.30	-7.0389739	113.9001919	4.4km
001	15.14.50	-7.032920	113.890290	5.7km
002	15.17.08	-7.028128	113.881482	6.9km
001	14.20.57	-7.017139	113.879313	8.1km
002	14.25.01	-7.005676	113.861654	10.3km

Tabel III. Data hasil pengukuran transmit data dengan 2 node

ID	Waktu	Latitude	Longitude	Jarak
001	14.53.20	-7.048762	113.934385	0m
002	14.56.50	-7.0469967	113.9300747	800m
003	14.59.10	-7.043621	113..923277	1.5km
001	15.03.40	-7.0371341	113.9104658	3.1km
002	15.06.13	-7.033686	113.907752	3.9km
003	15.10.30	-7.0389739	113.9001919	4.4km
001	15.14.50	-7.032920	113.890290	5.7km
002	15.17.08	-7.028128	113.881482	6.9km
003	14.20.57	-7.017139	113.879313	8.1km
003	14.25.01	-7.005676	113.861654	10.3km

### B. Receive Data Gateway

Setelah data di transmit oleh ketiga node, selanjutnya data akan diterima oleh gateway dan di

direct ke google *cloud* firebase untuk selanjutnya di kelola di backend dan ditampilkan di frontend. Dalam pengujian kali ini gateway diletakkan di rumah dengan ketinggian sekitar 7 meter diatas permukaan tanah. Frekuensi yang digunakan oleh gateway dan node sama-sama menggunakan alokasi frekuensi di 923 MHz. Setiap node mempunyai id unik tersendiri untuk membedakan antara node 1 dengan node lainnya. Berikut adalah tabel hasil dari penerimaan data dari ketiga node tersebut :

Tabel IV. Data hasil receive dengan 1 node

ID	Waktu	Latitude	Longitude	Jarak	RS SI	Delay	Lateny
001	14.53.20	-7.048762	113.934385	0m	-48	1s	0ms
001	14.56.50	-7.0469967	113.9300747	800m	-47	1s	0ms
001	14.59.10	-7.043621	113..923277	1.5km	-49	1s	0ms
001	15.03.40	-7.0371341	113.9104658	3.1km	-54	1s	0ms
001	15.06.13	-7.033686	113.907752	3.9km	-50	1s	0ms
001	15.10.30	-7.0389739	113.9001919	4.4km	-50	1s	0ms
001	15.14.50	-7.032920	113.890290	5.7km	-51	1s	20ms
001	15.17.08	-7.028128	113.881482	6.9km	-60	1s	200ms
001	14.20.57	-7.017139	113.879313	8.1km	-63	1s	500ms
001	14.25.01	-7.005676	113.861654	10.3km	-62	1s	500ms

Tabel VI. Data hasil pengukuran sumbu y- sejauh 5mm

Hasil pengukuran (mm)	Akurasi (%)	Error (%)
5.2	98	2,00
5.5	95	5,00
5.2	98	2,00
5	100	0,00
5	100	0,00
5.1	99	1,00
4.7	97	3,00
4.9	99	1,00
5.1	99	1,00
5.2	98	2,00
Rata-rata		1,7

Tabel V. Data hasil receive dengan 2 node

ID	Waktu	Latitude	Longitude	Jarak	RS SI	Delay	Lateny
001	14.53.20	-7.048762	113.934385	0m	-54	1s	0ms
002	14.56.50	-7.0469967	113.9300747	800m	-50	1s	0ms
001	14.59.10	-7.043621	113..923277	1.5km	-49	1s	0ms



00 2	15.0 3.40	- 7.0371 341	113.910 4658	3.1k m	-54	1s	100 ms
00 1	15.0 6.13	- 7.0336 86	113.907 752	3.9k m	-60	1s	200 ms
00 2	15.1 0.30	- 7.0553 12	113.909 723	4.4k m	-62	1s	200 ms
00 1	15.1 4.50	- 7.00* %8	113.09\$ #@27	5.7k m	-61	1s	200 ms
00 2	15.1 7.08	- 7.033.- #9	113.9- #@#\$2	6.9k m	-62	1s	500 ms
00 1	14.2 0.57	- 7.050+ @!#	113.09\$ #@7	8.1k m	-62	1s	500 ms
00 2	14.2 5.01	- 7.05)* &#@	113.9%# *3	10.3 km	-61	1s	500 ms

Tabel VI. Data hasil receive dengan 3 node

I D	Waktu	Latitude	Longitude	Jarak	RSI	Delay	Latency
00 0 3	14.53 .20	- 7.0487 62	113.934 385	0m	-64	1s	50ms
00 0 2	14.56 .50	- 7.0469 967	113.930 0747	800 m	-60	1s	50ms
00 0 3	14.59 .10	- 7.0436 2199	113.92 3277	1.5k m	-62	1s	50ms
00 0 2	15.03 .40	- 7.0371 341	113.910 4658	3.1k m	-68	1s	100 ms
00 0 1	15.06 .13	- 7.0336 86	113.907 752	3.9k m	-10	1s	500 ms
00 0 2	15.10 .30	- 7.0553 12	113.909 723	4.4k m	-12	1s	200 ms
00 0 1	15.14 .50	- 7.0509 38	113.099 827	5.7k m	-90	1s	500 ms
00 0 2	15.17 .08	- 7.033.- #9	113.9- #@#\$2	6.9k m	-12	1s	500 ms
00 0 1	14.20 .57	- 7.050+ @!#	113.09\$ #@7	8.1k m	-15	1s	1000 ms
00 0 2	14.25 .01	- 7.05)* &#@	113.9%# *3	10.3 km	-20	1s	1000 ms

Berdasarkan data dari ketiga tabel di atas yang merupakan data rekam dari 3 kondisi yang berbeda. Yang pertama node 1 aktif, node 2 dan 3 nonaktif, yang kedua node 1, 2 aktif dan node 3 nonaktif serta yang ketiga semua node aktif. Dalam kondisi aktif semua node mengirimkan data bersamaan dan diterima oleh gateway lalu dikirimkan ke cloud secara bersamaan. Dari ketiga hasil tersebut terlihat ketika hanya node 1 yang aktif, jarak transmisi yang dapat di terima sejauh

5.7 km dengan RSSI rata-rata -48 – (-53) dan latency baru terjadi di jarak 6.9 km . Untuk selanjutnya 2 node aktif, jarak transmisi yang dapat diterima sejauh 4.4 km dengan RSSI rata-rata -54 –(-62) dan latency terjadi di jarak 3,1km namun kecil, latency mulai membesar di jarak 6.9 km. Selanjutnya yang terakhir 3 node aktif, jarak transmisi yang dapat diterima sejauh 4.4 km. Ini sama dengan ketika 2 node aktif, namun perbedaan yang signifikan terletak di besar RSSI serta latency nya. Untuk RSSI rata-rata sebesar -80 – (-150) dan latency cenderung lebih besar daripada kedua kondisi yang pertama. Latency terjadi sejak jarak 0 km namun kecil, dan membesar di jarak 6.9 km. Dari ketiga data tersebut dapat disimpulkan jika semakin bertambahnya node yang mentransmisikan data maka latency dan RSSI akan semakin besar. Hal ini dikarenakan ketiga node mengirimkan data kepada gateway yang sama.

### C. Pengujian Aplikasi

Pengujian kali ini dilakukan untuk menguji apakah software yang telah dibuat telah berjalan sebagaimana mestinya atau tidak. Ada beberapa skema aplikasi yang perlu di uji coba diantara lain yaitu : menampilkan marker, perubahan warna marker, serta pengiriman himbauan langsung dari maps berupa sms.

#### 1. Menampilkan gambae pada Maps

##### a) Kirimkan data ke google cloud firebase

CD <https://cobaapi-default-rtdb.firebaseio.com/data/>

```
cobaapi-default-rtdb > data > 1
1
-- datatry: 950131
-- lat: 0
-- latitude2: -7.08
-- lon: 0
-- longitude2: 1139345
-- latitude: 0
-- longitude: 0
```

Gambar 5. Tampilan Firebase

##### b) Parsing data dari cloud menggunakan REST API lalu implementasikan di php

```
<?php
include ('../includes/dbconfig.php');
include "koneksi.php";

$gps="data/";
$fetch = $database->getReference($gps)->getValue();
foreach($fetch as $key => $row)

$lat=$row["latitude"];
$lon=$row["longitude"];
```

Gambar 6. Code parse data dari Firebase menggunakan REST API

##### c) Implementasikan data yang telah di parse ke dalam maps

```
map->addControl(new mapboxgl.NavigationControl());
//marker pasien dan pemberian warna dari hasil penentuan jarak aman
var marker = new mapboxgl.Marker({color: "purple"});
$.getJSON('http://api.geoip.net/v2/json/lat?lat=' + lat + '&apikey=' + apiKey, function(data) {
    $.each(data, function(index, value) {
        //setLatLng( $lat$ ,  $lon$ );
        //setLngLat( $lon$ ,  $lat$ );
        //setPopup(new mapboxgl.Popup().setText( $lat$  + ' ' +  $lon$ ));
        var marker = new mapboxgl.Marker({color: "purple"});
        //setLatLng( $lat$ ,  $lon$ );
        //setLngLat( $lon$ ,  $lat$ );
        //setPopup(new mapboxgl.Popup().setText( $lat$  + ' ' +  $lon$ ));
        var marker = new mapboxgl.Marker({color: "purple"});
        //setLatLng( $lat$ ,  $lon$ );
        //setLngLat( $lon$ ,  $lat$ );
        //setPopup(new mapboxgl.Popup().setText( $lat$  + ' ' +  $lon$ ));
        var marker = new mapboxgl.Marker({color: "purple"});
        //setLatLng( $lat$ ,  $lon$ );
        //setLngLat( $lon$ ,  $lat$ );
        //setPopup(new mapboxgl.Popup().setText( $lat$  + ' ' +  $lon$ ));
        var marker = new mapboxgl.Marker({color: "purple"});
        //setLatLng( $lat$ ,  $lon$ );
        //setLngLat( $lon$ ,  $lat$ );
        //setPopup(new mapboxgl.Popup().setText( $lat$  + ' ' +  $lon$ ));
```

Gambar 7. Code implementasi data hasil parse

d) Tentukan warna marker berdasarkan kategori ODP, PDP.

```
//penentuan warna marker dari jarak dan status
if ($status1=="menjauh"){
echo $color="black";
} if ($status1=="aman" && $pasien=="odp"){
echo $color = 'yellow';
} if ($status1=="aman" && $pasien=="pdp"){
echo $color = 'red';
}
//metode lain penentuan jarak aman dan warna
if ($status2=="menjauh"){
echo $color2='black';
} if ($status2=="aman" && $pasien2=="odp"){
echo $color2 = 'yellow';
} if ($status2=="aman" && $pasien2=="pdp"){
echo $color2 = 'red';
}
if ($status3=="menjauh"){
echo $color3="black";
} if ($status3=="aman" && $pasien3=="odp"){
echo $color3 = 'yellow';
} if ($status3=="aman" && $pasien3=="pdp"){
echo $color3 = 'red';
}
}
```

Gambar 8. Code penentuan warna marker

e) Masukkan latitude dan longitude pengguna/pasien sebagai acuan dalam penentuan zona aman

NO	NAMA	ALAMAT	NO HP	LATITUDE	LONGITUDE	STATUS	AKSI
1	Moh Inaduddin	kamal kamal	087780528270	-7.1283217	112.7254235	odp	lihat
2	Abdillah Al-Ghifary	Kalianget Timur	087780528270	-7.133557	112.718089	odp	lihat
3	Affan Maulana	Gapura	087772000362	-7.047934	113.9311488	odp	lihat
4	hinatachan	sunu	08743526182	-7.937055	112.716304	pdp	lihat

Gambar 9. Data Pasien

## V. KESIMPULAN

Hal-hal yang dapat disimpulkan dalam penelitian kali ini ialah :

1. LoRaWAN dapat menjadi alternatif teknologi IoT baru terutama di daerah perkotaan ataupun pedesaan yang minim akan sinyal internet ataupun seluler.
2. Dengan teknologi LoRaWAN pemakaian bandwidth diantara semua node disamakan, ini bertujuan agar data yang dikirimkan dapat disampaikan secara bersamaan ke gateway.
3. Penelitian ini menggunakan Gateway dengan spesifikasi ketinggian +- 7m diatas permukaan tanah, frekuensi 923MHz dengan modul LoRa sx1276. Dengan 3 node mentransmit data secara bersamaan gateway ini mampu menerima jarak optimal kurang lebih sejauh 4 kilometer. Dengan RSSI terbesar di kisaran -200.
4. Node yang digunakan untuk mentransmisikan data menggunakan wemos d1 mini sebagai mikro utama, gps neo 6 serta sx1276. Sedangkan baterai yang digunakan bertipe Li-Po 3.7v 200 mA.H.
5. Hasil pengujian cloud menggunakan google cloud firebase, firebase dapat menerima data realtime dengan baik dan akurat. Serta mempunyai fitur filter karakter-karakter selain angka dan string.
6. Mapbox API digunakan sebagai interface maps menggunakan aplikasi web, untuk fitur-fitur yang digunakan antara lain marker, warna marker, pencarian, dialog box dan zoom out, zoo in.
7. Selain API Mapbox penelitian ini juga menggunakan API dari Firebase yang digunakan untuk memarsing data dari realtime firebase serta API dari covid19.mathro.id sebagai pemantauan perkembangan covid secara realtime di indonesia.

8. Metode heversine digunakan untuk mengkalkulasi jarak antara 2 buah titik kordinat yang berbeda. Hal ini digunakan untuk mengidentifikasi jika terjadi pelanggaran zona aman isolasi mandiri. Heversine digunakan karena memiliki selisih yang relatif kecil dengan jarak aslinya yaitu sekitar 2 – 3 meter.
9. PHP merupakan bahasa pemrograman backend yang paling banyak digunakan dalam penelitian ini.
10. QOS (Quality of Service) sangat terpengaruh oleh jarak node dalam mentransmisikan data dan jumlah node yang mentransmisikan data.

## DAFTAR PUSTAKA

- [1] R. Angriawan and N. Anugraha, "Sistem Pelacak Lokasi Sapi dengan Sistem Komunikasi LoRa," *Inspir. J. Teknol. Inf. dan Komun.*, vol. 9, no. 1, p. 33, 2019, doi: 10.35585/inspir.v9i1.2494.
- [2] W. P. Putra, R. I. M, A. Sumarudin, and A. E. Putro, "Implementasi Lorawan Server Untuk Sistem Tracking Perahu Nelayan Berbasis MQTT Protocol," *J. Appl. Informatics Comput.*, vol. 2, no. 2, pp. 46–50, 2018, doi: 10.30871/jaic.v2i2.1024.
- [3] A. Augustin, J. Yi, T. Clausen, and W. M. Townsley, "A study of Lora: Long range & low power networks for the internet of things," *Sensors (Switzerland)*, vol. 16, no. 9, pp. 1–18, 2016, doi: 10.3390/s16091466.
- [4] P. Devi, D. Istianti, S. Y. Prawiro, N. Bogi, A. Karna, and I. A. Nursafa, "Analisis Performansi Teknologi Akses LPWAN LoRa Antares Untuk Komunikasi Data End Node," *Citee 2019*, pp. 22–26, 2019.
- [5] E. D. Widianto, A. A. Faizal, D. Eridani, R. D. O. Augustinus, and M. S. Pakpahan, "Simple LoRa Protocol: Protokol Komunikasi LoRa Untuk Sistem Pemantauan Multisensor," *TELKA - Telekomun. Elektron. Komputasi dan Kontrol*, vol. 5, no. 2, pp. 83–92, 2019, doi: 10.15575/telka.v5n2.83-92.
- [6] J. Chen, K. Hu, Q. Wang, Y. Sun, Z. Shi, and S. He, "Narrowband Internet of Things: Implementations and Applications," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2309–2314, 2017, doi: 10.1109/JIOT.2017.2764475.
- [7] E. Winarno, W. Hadikurniawati, and R. N. Rosso, "Location based service for presence system using haversine method," *Proc. - 2017 Int. Conf. Innov. Creat. Inf. Technol. Comput. Intell. IoT, ICITech 2017*, vol. 2018-Janua, pp. 1–4, 2018, doi: 10.1109/INNOCIT.2017.8319153.
- [8] J. A. D. M. Bastos, L. M. Afonso, and C. S. De Souza, "Metacommunication between programmers through an application programming interface: A semiotic analysis of date and time APIs," *Proc. IEEE Symp. Vis. Lang. Human-Centric Comput. VL/HCC*, vol. 2017-October, pp. 213–221, 2017, doi: 10.1109/VLHCC.2017.8103470.