

## IMPLEMENTASI KONSEP IOT PADA PANIC BUTTON DENGAN MENGGUNAKAN GOOGLE FIREBASE

### IMPLEMENTATION OF IOT CONCEPT ON PANIC BUTTON USING GOOGLE FIREBASE

Luqmanul Hakim Muslim<sup>1</sup>, Rendy Munadi<sup>2</sup>, Arif Indra Irawan<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

<sup>1</sup>luqmanulhakimmuslim@student.telkomuniversity.ac.id,

<sup>2</sup>Rendymunadi@telkomuniversity.ac.id, <sup>3</sup>Arifirawan@telkomuniversity.ac.id

#### Abstrak

Untuk meningkatkan pelayanan di bidang keamanan bagi warga Indonesia khususnya tentang pelaporan suatu tindak kriminal, diperlukan adanya teknologi pendukung yang dapat dengan mudah dan cepat menghubungkan antar pelapor dengan pihak yang berwenang, sehingga dapat ditindak lanjuti dengan cepat dan tepat. Sehingga dibuat alat panic button yang mengimplementasikan konsep IoT untuk memudahkan pelaporan tindak kriminal yang mengirimkan koordinat GPS melalui BOT Telegram dengan koordinat yang tersimpan secara realtime pada *realtime database* Gooogle Firebase. Data yang sudah dapat dikirim lalu dianalisis QoS dan akurasi GPS maka didapatkan data pergeseran lokasi rata rata dari koordinat sebenarnya sebesar 5,37 meter, *delay end to end* pada pengiriman pesan ke BOT Telegram dari NodeMCU didapatkan hasil paling besar 0,143 S dan paling kecil yaitu 0,087 S, *Throughput end to end* terbesar 2,304 kbps dan *throughput* terkecil yaitu 1,009 kbps bps dengan rata-rata 1,4477 kbps. *Jitter* terkecil yang didapat adalah 5,94421 ms, *jitter* terbesar 9,9086 ms dengan *jitter* rata-rata 7,8282 ms, semua parameter yang diuji sudah termasuk kedalam kategori baik menurut standarisasi TIPHON

**Kata Kunci : IoT, Google Firebase, Bot API**

#### Abstract

*To improve services in the field of security for Indonesian people, especially regarding reporting a crime, it is necessary to have supporting technology that can easily and quickly connect between the user and the authorities, so report can be followed up quickly and appropriately. Panic button device was created to implements the IoT concept to facilitate reporting of criminal acts that send GPS coordinates via Telegram BOT with coordinates stored to realtime database of the Gooogle Firebase. There is parameter tested, QoS and GPS accuracy shows that the average location shift data from the actual coordinates is 5.37 meters, the end-to-end delay in sending messages to the Telegram BOT from NodeMCU obtained the largest results of 0.143 S and the smallest which is 0.087 S, the largest throughput end to end is 2.304 kbps and the smallest throughput is 1.009 bps and the average is 1.4477 kbps. The lowest jitter end to end is 5.94421 ms and the largest jitter is 9.9086 ms with the average is 7.8282 ms. According to the standardization of TIPHON parameter is very good.*

**Keywords: IoT, Google Firebase, Bot API**

#### 1. Pendahuluan

Tingkat kriminalitas di Indonesia setiap tahunnya selalu mengalami perubahan. Data BSN menunjukkan tingkat kriminalitas mengalami kenaikan persentase, jumlah korban kekerasan dari 1,01 % di tahun 2018 menjadi 1.11% pada tahun 2019 dan angka pelaporan korban kekerasan sebesar 46,40%, dari data yang diperoleh menunjukkan korban kekerasan dan tindakan kriminal belum seluruhnya dapat melaporkan kepada pihak berwenang dikarenakan terkendala oleh beberapa faktor, salah satunya adalah kurang adanya fasilitas untuk menyampaikan laporan[1].

Berbeda dengan *panic button* yang sudah ada sebelumnya, *panic button* yang dirancang kali ini lebih mengutamakan kemudahan penggunaan dengan bentuk fisik yang mudah digunakan. *Panic button* yang sudah lebih dulu ada berbentuk aplikasi sehingga tidak memungkinkan dimiliki semua kalangan. Hal ini menjadi penghambat pelaporan[2]. *Panic Button* lainnya berbentuk aplikasi yang hanya terhubung ke suatu *website* sehingga cukup sulit untuk mengaksesnya, dan terlalu memakan waktu yang lama untuk mengoperasikannya dan tidak dilengkapi dengan database[3].

Penelitian ini menggunakan NodeMCU sebagai perangkat keras *panic button*, Google Firebase sebagai *database* yang dituju oleh *panic button* dan menyimpan data dari *panic button* dan akan menampilkan koordinat terbaru dari pengguna ketika dipanggil oleh BOT Telegram, dan memiliki kelebihan dapat menyimpan koordinat terakhir meskipun alat sudah tidak terkoneksi dengan internet. Telegram dipilih karena salah satu media sosial yang sifatnya masih *open source*. Google Firebase dipilih karena salah satu *cloud computing* yang memiliki fitur *realtime database*. *Cloud computing* dipilih karena cara baru untuk mengakses informasi, memproses, dan menganalisa data yang memudahkan manusia untuk mengakses data dari seluruh penjuru dunia[4].

## 2. Dasar Teori

### 2.1 Internet of Things (IoT)

IoT adalah suatu teknologi di masa depan mengenai telekomunikasi dan *computing* dimana semuanya berasal dari suatu konsep dimanapun dan kapan pun terkoneksi dengan segalanya. Unsur unsur yang wajib ada pada IoT adalah sensor, konektivitas dan perangkat berukuran kecil.

### 2.3 Cloud Computing

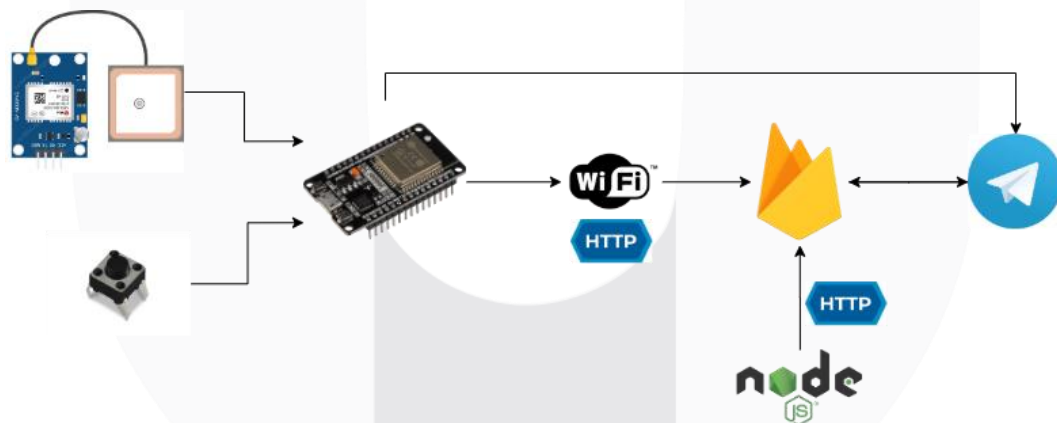
*Cloud Computing* merupakan suatu layanan dinamis yang mendistribusikan pelayanannya sebaik mungkin dengan mengurangi pengurangan *owner* dan sumber penyimpanan yang lebih baik. *Cloud computing* yang digunakan kali ini menggunakan *realtime database* Google Firebase yang dapat menyimpan data secara *realtime*.

### 2.4 Telegram

Telegram dipilih karena merupakan social media yang masih bersifat *open source* dan mendukung *webhook API*. Telegram BOT difungsikan sebagai media sosial pengirim dan penerima data dari *user*.

## 3. Desain Sistem

Desain sistem *panic button* terdiri dari beberapa perangkat keras yaitu NodeMCU sebagai mikrokontroler, modul GPS UBLOX NEO-6M, dan *push button* sebagai *trigger*. Perangkat lunak yang digunakan yaitu BOT Telegram untuk menerima data yang dihubungkan oleh NodeJS sebagai *backendnya*.

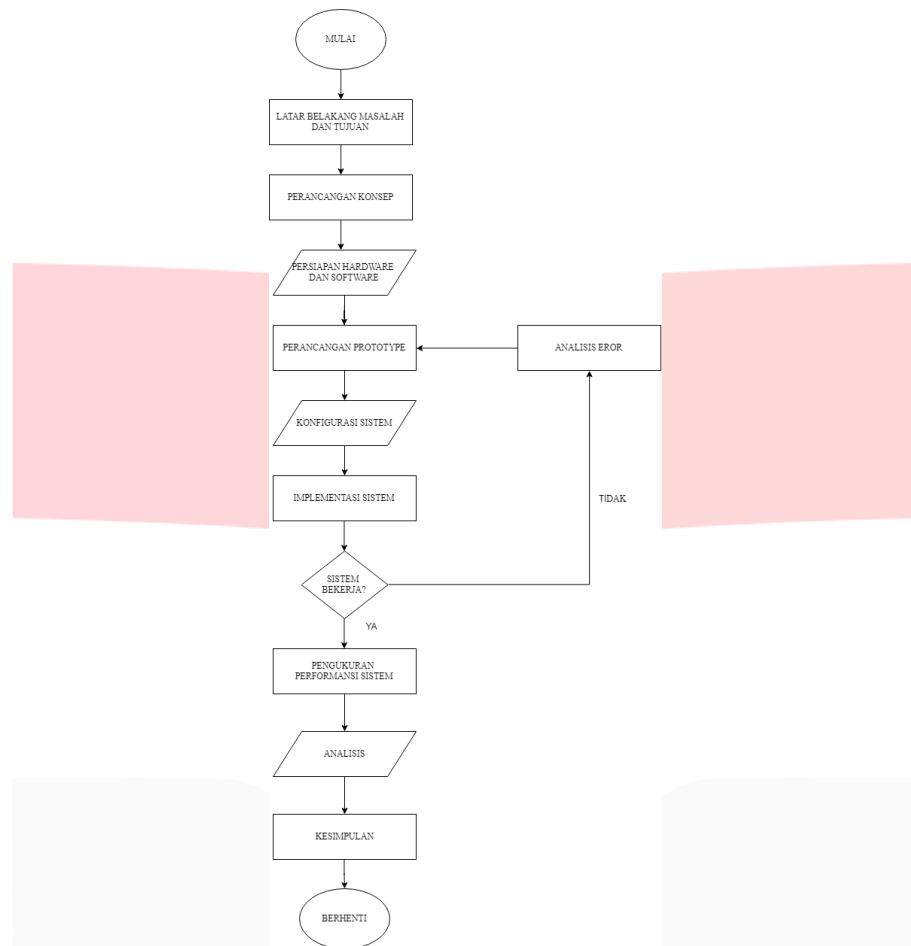


Gambar 1 Desain Sistem

Perancangan sistem kali ini dibagi menjadi 2 bagian yaitu mikrokontroler dan internet. Dimana mikrokontroler yang telah terpasang modul GPS akan mengirimkan koordinat ke *realtime database* dan menyimpannya, dan mengirimkan notifikasi bahaya ke BOT telegram untuk memberitahu kondisi korban kepada pihak yang berwenang.

### 3.2 Diagram Alir Utama

Langkah-langkah sistematis dalam melaksanakan penelitian Tugas Akhir ini dijelaskan pada Gambar 3



Gambar 2 Diagram alir utama

#### 4. Pengujian dan Analisis

Pada pengujian alat *panic button* dilakukan beberapa skenario pengujian untuk mengetahui performa dari alat yang telah dibuat, Adapun parameter yang digunakan pada pengujian performansi diharapkan dapat memenuhi syarat kriteria baik sehingga alat dapat beroperasi dengan lebih optimal. Parameter Tersebut diantaranya berupa *Delay*, *Throughput*, akurasi GPS dan analisis penggunaan *realtime database* Firebase. Parameter pengujian berdasarkan standarisasi TIPHON sebagai nilai acuan baik tidaknya parameter yang di uji, berikut ini beberapa standarisasi yang dibuat oleh TIPHON.

Tabel 1 Standarisasi delay TIPHON[5]

| Kategori Delay       | Besar Delay  |
|----------------------|--------------|
| <i>Excellent</i>     | <150 ms      |
| <i>Good</i>          | 150 – 300 ms |
| <i>Poor</i>          | 300 – 450 ms |
| <i>Unnacceptable</i> | >450 ms      |

Tabel 2 Standarisasi *Throughput* TIPHON[5]

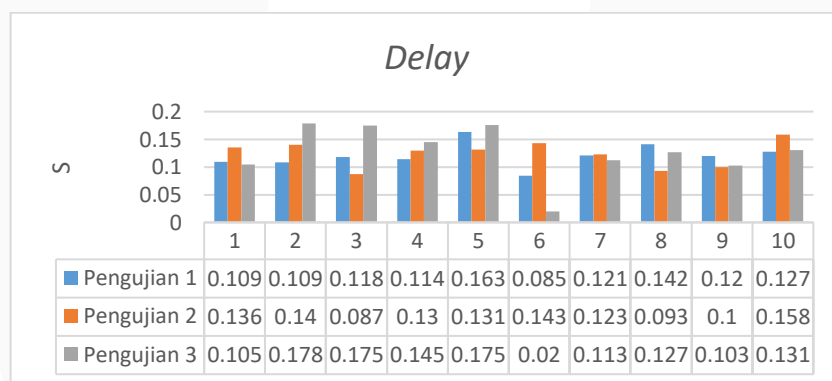
| Kategori <i>Throughput</i> | Besar <i>Throughput</i><br>(bps) |
|----------------------------|----------------------------------|
| <i>Excellent</i>           | 100                              |
| <i>Good</i>                | 75                               |
| <i>Poor</i>                | 50                               |
| <i>Unnacceptable</i>       | >25                              |

Tabel 3 Standarisasi *Jitter* TIPHON[5]

| Kategori <i>Jitter</i> | Peak <i>Jitter</i> (ms) |
|------------------------|-------------------------|
| <i>Excellent</i>       | 0                       |
| <i>Good</i>            | 0 - 75                  |
| <i>Poor</i>            | 5 - 125                 |
| <i>Unnacceptable</i>   | 125 - 225               |

#### 4.1 Delay end to end

Setelah dilakukan 30 kali pengujian *delay end to end* didapatkan hasil pada Gambar 3

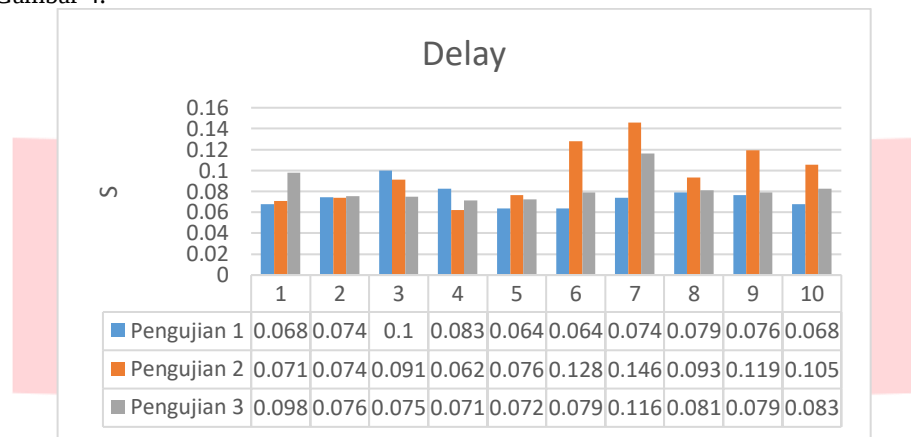


Gambar 3 Delay End to End

Pada Gambar 3 merupakan diagram rata - rata nilai *delay* tiap jarak. Pada grafik tersebut dapat dilihat *delay* terbesar yaitu mencapai 0,1783 S, *delay* terkecil yaitu 0,093 S dan *delay* rata-ratanya adalah 0,1239 S. Pada pengujian tersebut perubahan *delay* tidak terlalu signifikan, sehingga jarak tidak terlalu mempengaruhi terhadap besarnya nilai *delay*. *Delay* tersebut lebih besar dari *delay* pengiriman BOT Telegram seperti pada Gambar 4.3 dikarenakan ketika mengirim pesan ke telegram, NodeMCU mengirim koordinat ke Firebase sehingga *delay* yang didapatkan seperti pada Gambar 4.2, sehingga lama proses pengiriman data sesudah terhubung dengan Firebase akan semakin lama, namun hasil tersebut masih dalam titik kategori yang baik menurut standarisasi TIPHON.

#### 4.2 Delay NodeMCU – BOT Telegram

Dilakukan pengujian *delay* antara NodeMCU ke BOT Telegram sebanyak 30 kali, didapatkan hasil pada Gambar 4.

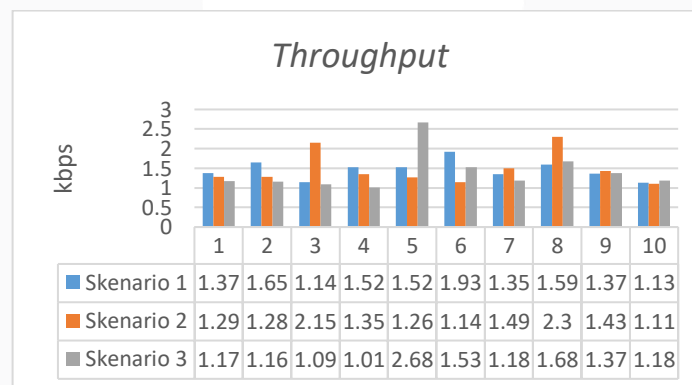


Gambar 4 Delay NodeMCU-Telegram

Pada Gambar 6 dilakukan pengambilan data setiap 3 kali percobaan. Hasil yang didapatkan dengan *delay* terbesar 0,1277 ms, *delay* terkecil yaitu 0,0622 ms dan *delay* rata-ratanya adalah 0,0848 ms. Pada pengujian tersebut perubahan *delay* tidak terlalu signifikan, sehingga Telegram tidak terlalu mempengaruhi terhadap besarnya nilai *delay* total. Hasil *delay* tersebut dikategorikan sangat bagus menurut standarisasi TIPHON.

#### 4.3 Rata-rata throughput end to end

Dilakukan 30 kali pengujian untuk mengukur nilai *throughput* seperti yang didapatkan pada Gambar 5

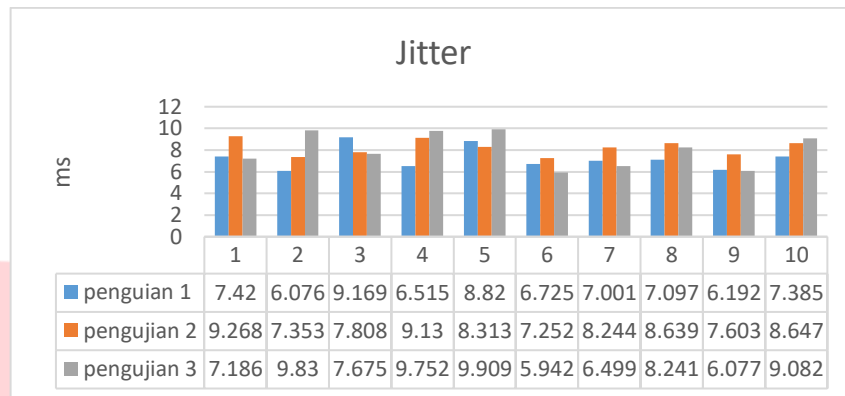


Gambar 5 Throughput end to end

Pada Gambar 7 dapat dilihat *throughput* terbesar yaitu mencapai 2,304 kbps, *throughput* terkecil yaitu 1,009 kbps dan *throughput* rata-ratanya adalah 1,44778 kbps. Nilai ini diperoleh sangat baik, dan data yang dikategorikan Besar menurut standarisasi TIPHON sehingga nilai *throughput* yang didapat sangat baik.

#### 4.4 Rata-rata jitter end to end

Dilakukan 30 kali pengujian untuk mengukur nilai *jitter* seperti yang didapatkan pada Gambar 6



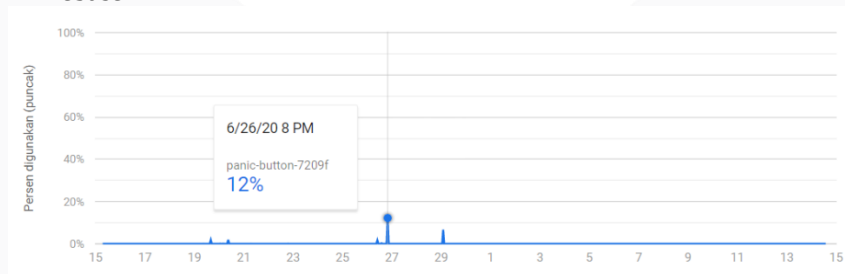
Gambar 6 Rata-rata Jitter

Pada grafik tersebut dapat dilihat *jitter* terbesar yaitu mencapai 9,8298 ms, *jitter* terkecil yaitu 5,9421 ms dan *jitter* rata-rata adalah 7,8282 ms. Nilai ini diperoleh termasuk kategori baik menurut standarisasi TIPHON karena berada pada rentang 0 s/d 75 ms.

#### 4.5 Memori usage Firebase

Penggunaan memori terbesar terdapat pada rentan waktu 4 hari terakhir dengan data tersimpan sebesar 42 byte dan data terkecil adalah 28 byte. Besarnya memori yang digunakan berdasarkan data yang diambil modeul GPS.

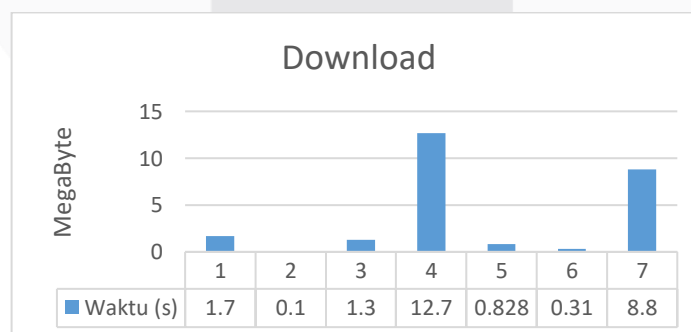
#### 4.6 Beban Firebase



Gambar 7 Beban Firebase

Pengamatan dilakukan selama 30 hari penggunaan, beban tertinggi yang diterima yaitu 12% setiap interval waktu 1 menit seperti yang terlihat gambar, dikarenakan intensitas penggunaan database yang intens untuk penghitungan QoS sebanyak 90 kali percobaan dalam kurun waktu yang setiap 30 detik percobaan .

#### 4.7 Download Firebase

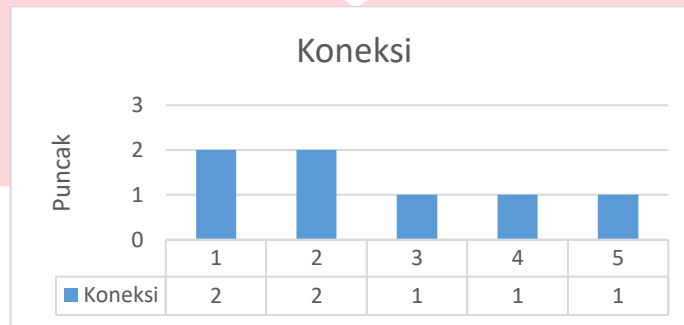


Gambar 8 Download Firebase

Pada Gambar 4.6 adalah total *byte* yang di *download* untuk semua *byte* yang di*download* dari *database* , termasuk *overhead* protokol koneksi dan enkripsi SSL, selama interval 1 menit yang digunakan selama 7 hari. Hasil *download* terbesar adalah pada hari ke 4 yaitu 12.7MB dan paling sedikit adalah 0,1MB. Beban tersebut merupakan proses kinerja dari *realtime database* untuk mengolah data selama 7 hari dilakukan pengujian.

#### 4.8 Koneksi Simultan Google Firebase

Pengujian koneksi simultan diambil sebanyak 5 kali pengujian dan didapatkan hasil seperti pada Gambar 7



*Gambar 9 Koneksi Simultan*

Koneksi simultan adalah banyaknya perangkat yang terkoneksi dengan Realtime database, dimana saya hanya menggunakan satu alat dan satu BOT Telegram sehingga nilai tertingginya adalah 2, Koneksi simultan dibatasi oleh layanan realtime database hanya sebesar 200.000 perangkat yang dapat terkoneksi.

#### 4.9 Pengujian Akurasi GPS

Pada pengujian koordinat dilakukan sebanyak 30 kali untuk mengukur perbedaan hasil koordinat yang diperoleh melalui GPS dan koordinat yang diambil manual dari google maps.

*Tabel 4 Akurasi GPS*

[illegible]

|           |           |         |            |          |
|-----------|-----------|---------|------------|----------|
| 107.711   | -6.905413 | 107.711 | -6.9056719 | 5.519021 |
| 107.711   | -6.905413 | 107.711 | -6.9056719 | 5.519021 |
| 107.711   | -6.905413 | 107.711 | -6.9056719 | 5.519021 |
| 107.711   | -6.905413 | 107.711 | -6.9056719 | 5.519021 |
| 107.711   | -6.905413 | 107.711 | -6.9056719 | 5.519021 |
| 107.711   | -6.905634 | 107.711 | -6.9056719 | 2.920626 |
| 107.7111  | -6.905618 | 107.711 | -6.9056719 | 3.323019 |
| 107.7111  | -6.905594 | 107.711 | -6.9056719 | 3.47867  |
| 107.7111  | -6.905594 | 107.711 | -6.9056719 | 3.47867  |
| 107.7111  | -6.905555 | 107.711 | -6.9056719 | 3.826554 |
| 107.7111  | -6.904126 | 107.711 | -6.9056719 | 28.44545 |
| 107.7111  | -6.905626 | 107.711 | -6.9056719 | 3.282753 |
| 107.7111  | -6.905638 | 107.711 | -6.9056719 | 3.233447 |
| 107.7111  | -6.905517 | 107.711 | -6.9056719 | 4.253833 |
| 107.7111  | -6.90538  | 107.711 | -6.9056719 | 6.209701 |
| Rata rata |           |         |            | 5.378794 |

Dari hasil pengujian diatas, pengujian koordinat lokasi dari GPS berselisih rata-rata 5.37 meter dari koordinat asli yang sudah ditentukan.

## 5. Simpulan

Alat *panic button* yang telah dibuat terdiri dari perangkat keras NodeMCU, modul GPS NEO Ublox 6M, *push button*, dan perangkat lunak BOT Telegram, *Backend NodeJS* dan Realtime database Firebase dapat berfungsi dengan baik mengirimkan pesan dan koordinat dengan *delay end to end* hasil paling besar 0,1277 S, *delay* terkecil yaitu 0,0622 S dan *delay* rata-ratanya adalah 0,0848 S, sedangkan *delay* NodeMCU-BOT Telegram terbesar yaitu 0,1277 S, *delay* terkecil yaitu 0,0622 S dan *delay* rata-ratanya adalah 0,0848 S, sehingga semua *delay* yang telah diuji termasuk kedalam kategori sangat bagus menurut standarisasi TIPHON. *Throughput End to End* terbesar 2,304 kbps, *throughput* terkecil yaitu 1,009 kbps dan *throughput* rata-ratanya adalah 1,44778 kbps *jitter* terbesar yaitu mencapai 9,8298 ms, *jitter* terkecil yaitu 5,9421 ms dan *jitter* rata-rata adalah 7,8282 ms sehingga termasuk kedalam kategori *throughput* yang sangat bagus dan *jitter* yang bagus menurut standarisasi TIPHON. Akurasi GPS UBLOX NEO-6m yang memiliki selisih rata rata untuk setiap koordinat adalah 5.37 meter.

## Daftar Pustaka:

- [1] B. P. Statistik, "Cover statistik kriminal 2019," *Cover Stat. Krim. 2019*, 2019.
- [2] M. Anike and M. Anike, "APPLICATION EMERGENCY PANIC BUTTON (AEPB) BERBASIS ANDROID (Studi Kasus RS St. Carolus Boromeus-Bello)," *Sistemasi*, vol. 8, no. 3, p. 367, 2019, doi: 10.32520/stmsi.v8i3.499.
- [3] A. I. Afolabi, O. Moses, M. S. Opeyemi, B.-O. A. Abraham, and A.-Z. P. Temidayo, "Design and Construction of A Panic Button Alarm System for Security Emergencies," *Int. J. Eng. Tech.*, vol. 4, no. 3, pp. 3–7, 2018, doi: 10.29126/23951303/IJET-V4I3P105.
- [4] A. dan W. Komputer, *Kupas Tuntas Berbagai Aplikasi Generasi Cloud Computing*. Semarang: ANDI, 2011.
- [5] Y. A. Pranata, I. Fibriani, and S. B. Utomo, "Analisis Optimasi Kinerja Quality of Service Pada Layanan Komunikasi Data Menggunakan Ns-2 Di Pt. Pln (Persero) Jember," *Sinergi*, vol. 20, no. 2, p. 149, 2016, doi: 10.22441/sinergi.2016.2.009.