

UNIVERSITAS
Dinamika

**SISTEM KOMUNIKASI MULTINODE LORA DENGAN GATEWAY
LORAWAN DI AREA PESISIR PANTAI**

LAPORAN TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

UNIVERSITAS
Dinamika

Oleh:

THEODORUS VISSER INULIMANG

19410200001

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2024

**SISTEM KOMUNIKASI MULTINODE LORA DENGAN GATEWAY
LORAWAN DI AREA PESISIR PANTAI**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik**



Disusun Oleh:

Nama : Theodorus Visser Inulimang

NIM : 19410200001

Program : S1 (Strata Satu)

Jurusan : Teknik Komputer

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2024

TUGAS AKHIR

SISTEM KOMUNIKASI MULTINODE LORA DENGAN GATEWAY LORAWAN DI AREA PESISIR PANTAI

Dipersiapkan dan disusun oleh:

Theodorus Visser Inulimang

NIM: 19410200001

Telah diperiksa, dibahas, dan disetujui oleh Dewan Pembahas

Pada: 1 Juli 2024

Susunan Dewan Pembahas

Pembimbing:

I. Musayyanah, S.ST., M.T.

NIDN 0730069102

II. Pauladie Susanto, S.Kom., M.T.

NIDN 0729047501

Pembahas:

Heri Pratikno, M.T., MTCNA., MTCRE.

NIDN 0716117302

Digitally signed by
Musayyanah
DN: cn=Musayyanah,
o=Universitas Dinamika,
ou=S1 Teknik Komputer,
email=musayyanah@dinamika
.ac.id, c=ID
Date: 2024.07.30 09:30:19
+07'00'

cn=Pauladie Susanto, o=Universitas
Dinamika, ou=PS S1 Teknik Komputer,
email=pauladie@dinamika.ac.id, c=ID
2024.07.30 16:47:54 +07'00'

Digitally signed by Heri Pratikno, M.T.
DN: cn=Heri Pratikno, M.T.,
o=Universitas Dinamika, ou=S1 Teknik
Komputer, email=heri@dinamika.ac.id,
c=ID
Date: 2024.07.31 10:13:03 +07'00'
Adobe Acrobat version: 11.0.23

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar sarjana

Digitally signed by Anjik

Sukmaaji

Date: 2024.08.02

08:26:14 +07'00'

Dr. Anjik Sukmaaji, S.Kom., M.Eng.

NIDN: 0731057301

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



“Dalam setiap petualanganmu, kau menemukan kebijaksanaan dan pengalaman yang memperkaya hidupmu.”

- Theodorus Visser Inulimang-

UNIVERSITAS
Dinamika

Atas berkat Tuhan Yang Maha Esa, saya dedikasikan karya ini kepada keluarga saya tercinta, yang selalu memberikan dukungan dan doa tanpa henti. Juga, kepada teman-teman Teknik Komputer Angkatan 2019, terima kasih atas bantuan, doa, dan dukungan yang telah memberikan semangat dan inspirasi untuk terus belajar dan berkembang menjadi pribadi yang lebih baik.



UNIVERSITAS
Dinamika

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : Theodorus Visser Inulimang
NIM : 19410200001
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Tugas Akhir
Judul Karya : **SISTEM KOMUNIKASI MULTINODE LORA
DENGAN GATEWAY LORAWAN DI AREA PESISIR
PANTAI**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, 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 diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar keesarjaan yang telah diberikan kepada Saya.

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 1 Juli 2024



Theodorus Visser Inulimang
NIM : 19410200001

ABSTRAK

LoRa adalah teknologi komunikasi nirkabel yang menggunakan modulasi *Chirp Spread Spectrum* (CSS) dan dapat beroperasi pada frekuensi sub-gigahertz serta 2,4 GHz untuk kecepatan data yang lebih tinggi. Protokol LoRaWAN, yang dibangun dari modulasi LoRa pada layer *Media Access Control* (MAC), memungkinkan koneksi perangkat sensor atau aktuator ke internet melalui server. Beberapa *platform cloud* server yang kompatibel dengan LoRaWAN adalah *The Things Network*, wisDM, ChirpStack, *Amazon Web Services* (AWS), dan Azure. Pada Tugas Akhir ini, LoRaWAN dimanfaatkan untuk menerima data posisi dari perangkat *end node* yang dilengkapi dengan sensor GPS. Data GPS ini kemudian dihitung jaraknya menggunakan metode *Haversine*, dan menghasilkan *error* dibawah 5% dalam pengukuran jarak antara dua titik koordinat di area pesisir pantai kenjeran kota Surabaya dengan beberapa skenario pengujian. Selain itu, penelitian ini juga mengevaluasi performa komunikasi LoRaWAN dengan nilai RSSI pada pengujian pertama dengan rentang -51,33dBm hingga -105,67dBm, pengujian kedua dengan rentang -72,73dBm hingga -102,22dBm, dan pengujian ketiga dengan rentang -88,89 dBm hingga -100,14dBm. SNR pada pengujian pertama menghasilkan perubahan sinyal *noise* rentang 13,42dBm hingga -10,17dBm, pengujian kedua dengan rentang 0,22dBm hingga -14,83dBm, dan pengujian ketiga dengan rentang 2,94 hingga -1,94. *Spreading Factor* pada pengujian pertama dan kedua menghasilkan perubahan nilai dari 7 hingga 10, sedangkan pengujian ketiga menghasilkan perubahan nilai dari 9 hingga 10. Dan *Airtime* pada pengujian Pertama menghasilkan rentang nilai 0,07s sampai 0,37s, pengujian kedua menghasilkan rentang nilai *Airtime* 0,11s sampai 0,37s, dan pengujian ketiga menghasilkan rentang nilai *Airtime* 0,19s sampai 0,37s.

Kata Kunci: *Haversine*, LoRa, LoRaWAN, *The Things Network*.

KATA PENGANTAR

Puji syukur saya panjatkan pada Tuhan Yang Maha Esa atas segala rahmat yang telah diberikan-Nya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul “*Sistem Komunikasi Multinode Lora Dengan Gateway Lorawan Di Area Pesisir Pantai*”. Dalam perjalanan menyelesaikan pengerjaan Laporan Tugas Akhir ini penulis banyak mendapatkan bantuan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Tuhan Yang Maha Esa, karena dengan rahmat-Nya penulis dapat menyelesaikan Laporan Tugas Akhir ini.
2. Orang tua dan seluruh keluarga yang telah memberikan dorongan dan dukungan baik secara moral maupun materi, sehingga penulis dapat menempuh dan menyelesaikan Laporan Tugas Akhir ini.
3. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembahas. Penulis mengucapkan terimakasih atas bimbingan yang diberikan dan kesempatan serta tuntunan baik secara lisan maupun tertulis, sehingga penulis dapat menyelesaikan Tugas Akhir.
6. Ibu Musayyanah, S.ST., M.T. selaku dosen pembimbing yang telah memberikan dukungan berupa motivasi, wawasan, dan saran bagi penulis selama pelaksanaan pengerjaan Tugas Akhir dan dalam pembuatan laporan Tugas Akhir.
7. Bapak Pauladie Susanto, S.Kom., M.T., selaku dosen pembimbing yang banyak memberikan masukan dan solusi agar Tugas Akhir ini menjadi lebih baik dan penulis dapat menyelesaikan Tugas Akhir ini.
8. Seluruh rekan-rekan S1 Teknik Komputer angkatan 2019 yang telah memberikan dukungan dan semangatnya untuk membantu penulis menyelesaikan laporan Tugas Akhir ini.

9. Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya laporan Tugas Akhir ini.

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, 1 Juli 2024

Penulis



UNIVERSITAS
Dinamika

DAFTAR ISI

	Halaman
ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	3
BAB II LANDASAN TEORI	4
2.1 Infrastruktur Komunikasi di daerah Pesisir Pantai.....	4
2.2 LoRa	4
2.3 LoRa Gateway.....	7
2.4 LoRaWAN.....	8
2.5 <i>The Things Network</i> (TTN)	8
2.6 Mode Aktivasi <i>End Node</i> ke TNN	13
2.6.1 Activation By Personalization (ABP).....	13
2.6.2 Over The Air Activation (OTAA)	14
2.6.3 Perbedaan ABP dan OTAA	15
2.7 Datacake	16
2.8 Arduino Due	16
2.9 Global Positioning System (GPS)	18
2.10 Algoritma <i>Haversine</i>	19
BAB III METODOLOGI PENELITIAN	20
3.1 Model Perancangan	20
3.1.1 LoRa End Node.....	21
3.1.2 Gateway	22
3.1.3 Server The Things Network.....	23

3.1.3.1	Pemilihan <i>Cluster</i>	24
3.1.3.2	Mendaftarkan perangkat <i>End Node</i>	25
3.1.3.3	<i>Register End Node</i>	26
3.1.3.4	<i>Payload Formatter</i>	28
3.1.3.5	<i>Intergrations Datacake</i>	30
3.1.4	Datacake.....	35
3.1.4.1	<i>Add device</i>	36
3.1.4.2	<i>Device Configuration</i>	39
3.1.4.3	<i>Payload Decoder</i>	41
3.1.4.4	<i>Dahsboard Device</i>	46
3.2	Perancangan Perangkat Keras	56
3.2.1	Perancangan LoRa <i>End Node</i>	56
3.2.2	Rangkaian Skematik LoRa <i>End Node</i>	57
3.2.3	Perancangan <i>Gateway</i>	59
3.3	Perancangan Perangkat Lunak	59
3.3.1	Algoritma Penerapan Sistem LoRa <i>End Node</i>	60
3.3.2	Algoritma Penerapan Sistem Pada <i>Gateway</i>	62
3.3.3	Algoritma Penerapan Sistem Pada Server TTN.....	63
3.3.4	Algoritma Penerapan Sistem Pada <i>Datacake</i>	64
BAB IV HASIL DAN PEMBAHASAN.....		68
4.1	Pengujian Pengiriman Data <i>End Node</i> pada <i>Gateway</i> dan Server TTN	68
4.1.1	Tujuan.....	68
4.1.2	Peralatan yang Digunakan.....	68
4.1.3	Cara Pengujian	68
4.1.4	Skenario Pengujian Pertama.....	69
4.1.5	Hasil Pengujian.....	72
4.1.6	Analisis Data	81
4.2	Skenario Pengujian Kedua	82
4.2.1	Tujuan.....	82
4.2.2	Peralatan yang Digunakan.....	82
4.2.3	Cara Pengujian	82
4.2.4	Pengujian Komunikasi <i>Multi Node</i>	83
4.2.5	Hasil Pengujian.....	87
4.2.6	Analisis Data	95

4.3 Pengujian <i>End Node</i> Pada Kondisi Kondisi Pengunjung	96
4.3.1 Tujuan.....	96
4.3.2 Peralatan yang Digunakan.....	97
4.3.3 Cara Pengujian	97
4.3.4 Skenario Pengujian Ketiga	97
4.3.5 Hasil Pengujian.....	99
4.3.6 Analisis Data	105
BAB V KESIMPULAN	107
5.1 Kesimpulan.....	107
5.2 Saran	108
DAFTAR PUSTAKA.....	109
LAMPIRAN	111
BIODATA PENULIS.....	149



UNIVERSITAS
Dinamika

DAFTAR GAMBAR

	Halaman
Gambar 2. 1 Jangkauan komunikasi nirkabel	5
Gambar 2. 2 Data Frame LoRa	6
Gambar 2. 3 <i>The Things Network</i>	8
Gambar 2. 4 <i>Physical and Communication layers of a LoRaWAN Network</i>	9
Gambar 2. 5 <i>Activation By Personalization</i>	13
Gambar 2. 6 <i>Over The Air Activation</i>	14
Gambar 2. 7 <i>Datacake</i>	16
Gambar 2. 8 <i>Arduino Due Board</i>	16
Gambar 2. 9 GPS NEO-6M	18
Gambar 3. 1 Model Perancangan Sistem	20
Gambar 3. 2 Perbandingan GPS Garmin dan Metode <i>Haversine</i>	21
Gambar 3. 3 <i>Gateway</i>	22
Gambar 3. 4 <i>Dashboard Gateway</i>	23
Gambar 3. 5 <i>Pemilihan cluster TTN</i>	24
Gambar 3. 6 <i>TTN application</i>	25
Gambar 3. 7 <i>TTN Create Application</i>	25
Gambar 3. 8 <i>TTN End Decive List</i>	26
Gambar 3. 9 <i>TTN End Device Type</i>	26
Gambar 3. 10 <i>TTN Generate Join EUI</i>	27
Gambar 3. 11 <i>TTN Generate DevEui dan AppKey</i>	27
Gambar 3. 12 <i>TTN End Decive List</i>	28
Gambar 3. 13 <i>TTN End Node Payload Formatter</i>	28
Gambar 3. 14 <i>TTN End Device Formatter Code</i>	29
Gambar 3. 15 <i>TTN Application Add API KEYS</i>	30
Gambar 3. 16 <i>TTN Application API KEYS</i>	30
Gambar 3. 17 <i>TTN Application Edit API KEY</i>	31
Gambar 3. 18 <i>TTN Application Webhooks</i>	31
Gambar 3. 19 <i>TTN Application Add Webhooks</i>	31
Gambar 3. 20 <i>TTN Add Webhooks Datacake</i>	31

Gambar 3. 21 TTN Setup Webhook For Datacake	31
Gambar 3. 22 TTN Datacake API Token.....	31
Gambar 3. 23 TTN TTN Edit Webhook For Datacake	31
Gambar 3. 24 TTN TTN Webhook Enabled Event Type	31
Gambar 3. 25 Login Datacake	31
Gambar 3. 26 Datacake Devices	31
Gambar 3. 27 Datacake Add Device	31
Gambar 3. 28 Datacake Add Device Type	31
Gambar 3. 29 Datacake Add Device Step 1	31
Gambar 3. 30 Datacake Add Device Step 2	31
Gambar 3. 31 TTN End Device DevEUI	31
Gambar 3. 32 Datacake Add Device Step 3	31
Gambar 3. 33 Datacake Add Device Step 4	50
Gambar 3. 34 Datacake Device Configuration.....	50
Gambar 3. 35 Datacake Device Downlink Configuration	51
Gambar 3. 36 Datacake Device Downlink TTI Configuration	51
Gambar 3. 37 Datacake Device Downlink Configuration	52
Gambar 3. 38 Datacake Device payload decoder.....	52
Gambar 3. 39 Datacake Device payload decoder.....	53
Gambar 3. 40 Datacake Device Add Field.....	53
Gambar 3. 41 Datacake Device payload decoder.....	54
Gambar 3. 42 Datacake Device Type Field	54
Gambar 3. 43 Datacake Device List Field	55
Gambar 3. 44 Datacake Device Debug.....	55
Gambar 3. 45 Datacake Device Debug Log.....	56
Gambar 3. 46 Datacake Device Debug Payload Decoder.....	56
Gambar 3. 47 Datacake Device Debug JSON TTN	56
Gambar 3. 48 Datacake Device Dashboard	57
Gambar 3. 49 Datacake Device Enable Edit	57
Gambar 3. 50 Datacake Device Widget	58
Gambar 3. 51 Datacake Device Add Widget.....	58
Gambar 3. 52 Datacake Device Widget Type	59

Gambar 3. 53 <i>Datacake Device Widget Title</i>	59
Gambar 3. 54 <i>Datacake Device Widget Field</i>	60
Gambar 3. 55 <i>Datacake Device Widget</i>	60
Gambar 3. 56 <i>Datacake Device Widget</i>	61
Gambar 3. 57 <i>Datacake Device Public Link</i>	61
Gambar 3. 58 <i>Datacake Device Create Public Link</i>	62
Gambar 3. 59 <i>Datacake Device Public Link Mode</i>	62
Gambar 3. 60 <i>Datacake Device Save Public Link</i>	63
Gambar 3. 61 <i>Datacake Device Dashboard visual</i>	63
Gambar 3. 62 <i>Datacake Aplikasi Handphone</i>	64
Gambar 3. 63 <i>Add Device Datacake dari Aplikasi Handphone</i>	64
Gambar 3. 64 <i>Title Device Datacake dari Aplikasi Handphone</i>	65
Gambar 3. 65 <i>List Device Datacake Dari Aplikasi Handphone</i>	65
Gambar 3. 66 <i>Dashboard Device Datacake Dari Aplikasi Handphone</i>	66
Gambar 3. 67 <i>Perancangan LoRa End Node</i>	67
Gambar 3. 68 <i>Skematik LoRa End Node</i>	68
Gambar 3. 69 <i>Perancangan Gateway ke Server TTN dan Datacake</i>	70
Gambar 3. 70 <i>Flowchart Sistem LoRa End Node</i>	71
Gambar 3. 71 <i>Tampilan request join Pada Serial monitor Handphone</i>	72
Gambar 3. 72 <i>Flowchart Sistem Pada Gateway</i>	73
Gambar 3. 73 <i>Flowchart Sistem Pada Datacake</i>	75
Gambar 4. 1 <i>Skenario Pengujian Pertama</i>	69
Gambar 4. 2 <i>Posisi Gateway Skenario Pengujian Pertama</i>	71
Gambar 4. 3 <i>Posisi Perangkat end node</i>	72
Gambar 4. 4 <i>Pengukuran Jarak Posisi 1 Google Earth</i>	72
Gambar 4. 5 <i>Pengukuran Jarak Posisi 2 Google Earth</i>	73
Gambar 4. 6 <i>Pengukuran Jarak Posisi 3 Google Earth</i>	74
Gambar 4. 7 <i>Pengukuran Jarak Posisi 4 Google Earth</i>	75
Gambar 4. 8 <i>Pengukuran Jarak Posisi 5 Google Earth</i>	76
Gambar 4. 9 <i>Pengukuran Jarak Posisi 6 Google Earth</i>	76
Gambar 4. 10 <i>Pengukuran Jarak Posisi 7 Google Earth</i>	77
Gambar 4. 11 <i>Pengukuran Jarak Posisi 8 Google Earth</i>	78

Gambar 4. 12	Pengukuran Jarak Posisi 9 <i>Google Earth</i>	79
Gambar 4. 13	Pengukuran Jarak Posisi 10 <i>Google Earth</i>	80
Gambar 4. 14	Skenario Pengujian Kedua	83
Gambar 4. 15	Skenario Pengujian Kedua Perangkat <i>end node 2</i>	84
Gambar 4. 16	Posisi <i>Gateway Skenario Pengujian Kedua</i>	85
Gambar 4. 17	Posisi Perangkat <i>End Node 1</i>	86
Gambar 4. 18	Posisi Perangkat <i>End Node 2</i>	86
Gambar 4. 19	Posisi Perangkat <i>End Node 3</i>	87
Gambar 4. 20	Pengukuran Jarak <i>end node 1</i> dengan <i>Google Earth</i>	87
Gambar 4. 21	Jarak <i>end node 2</i> Sesi 1 dengan GPS Garmin	89
Gambar 4. 22	Jarak <i>end node 2</i> Sesi 2 dengan GPS Garmin	90
Gambar 4. 23	Jarak <i>end node 2</i> Sesi 3 dengan GPS Garmin	91
Gambar 4. 24	Jarak <i>end node 2</i> Sesi 4 dengan GPS Garmin	92
Gambar 4. 25	Jarak <i>end node 2</i> Sesi 5 dengan GPS Garmin	93
Gambar 4. 26	Jarak <i>end node 3</i> dengan <i>Google Earth</i>	94
Gambar 4. 27	Hasil Visualisasi Pada <i>Dashboard Datacake</i>	95
Gambar 4. 28	Skenario Pengujian Ketiga	97
Gambar 4. 29	Posisi <i>Gateway Skenario Pengujian Ketiga</i>	98
Gambar 4. 30	Posisi Perangkat <i>End Node</i>	99
Gambar 4. 31	Jarak <i>end node</i> Sesi 1 dengan GPS Garmin	100
Gambar 4. 32	Jarak <i>end node</i> Sesi 2 dengan GPS Garmin	101
Gambar 4. 33	Jarak <i>end node</i> Sesi 3 dengan GPS Garmin	102
Gambar 4. 34	Jarak <i>end node</i> Sesi 4 dengan GPS Garmin	103
Gambar 4. 35	Jarak <i>end node</i> Sesi 5 dengan GPS Garmin	104

DAFTAR TABEL

	Halaman
Tabel 2. 1 Perbandingan komunikasi nirkabel.....	5
Tabel 2. 2 <i>List Frekuensi Plan</i>	10
Tabel 2. 3 Spesifikasi Kelas Perangkat LoRaWAN.....	11
Tabel 2. 4 Spesifikasi Arduino Due	17
Tabel 2. 5 Spesifikasi GPS NEO-6M.....	18
Tabel 3. 1 PIN Arduino Due Ke <i>Transmitter</i> Lora dan Sensor GPS	58
Tabel 4. 1 Jarak <i>Google Earth</i> Pengujian Pertama	69
Tabel 4. 2 Hasil Pengujian <i>end node</i> Posisi 1	73
Tabel 4. 3 Hasil Pengujian <i>end node</i> Posisi 2	73
Tabel 4. 4 Hasil Pengujian <i>end node</i> Posisi 3	74
Tabel 4. 5 Hasil Pengujian <i>end node</i> Posisi 4	75
Tabel 4. 6 Hasil Pengujian <i>end node</i> Posisi 5	76
Tabel 4. 7 Hasil Pengujian <i>end node</i> Posisi 6	77
Tabel 4. 8 Hasil Pengujian <i>end node</i> Posisi 7	77
Tabel 4. 9 Hasil Pengujian <i>end node</i> Posisi 8	78
Tabel 4. 10 Hasil Pengujian <i>end node</i> Posisi 9	79
Tabel 4. 11 Hasil Pengujian <i>end node</i> Posisi 10	80
Tabel 4. 12 Hasil Analisis Pengujian Skenario Pertama.....	81
Tabel 4. 13 Jarak <i>Google Earth</i> Pengujian Kedua.....	84
Tabel 4. 14 Jarak GPS Garmin Pengujian Kedua	84
Tabel 4. 15 Hasil Pengujian <i>end node</i> 1 Skenario 2	88
Tabel 4. 16 Hasil Pengujian <i>end node</i> 2 sesi 1	89
Tabel 4. 17 Hasil Pengujian <i>end node</i> 2 sesi 2.....	90
Tabel 4. 18 Hasil Pengujian <i>end node</i> 2 sesi 3.....	91
Tabel 4. 19 Hasil Pengujian <i>end node</i> 2 sesi 4.....	92
Tabel 4. 20 Hasil Pengujian <i>end node</i> 2 sesi 5.....	93
Tabel 4. 21 Hasil Pengujian <i>end node</i> 3 Skenario 2	94
Tabel 4. 22 Hasil Analisis Pengujian Skenario Kedua	95
Tabel 4. 23 Jarak GPS Garmin Pengujian Ketiga	98
Tabel 4. 24 Hasil Pengujian <i>end node</i> posisi Pertama	100

Tabel 4. 25 Hasil Pengujian <i>end node</i> posisi kedua.....	101
Tabel 4. 26 Hasil Pengujian <i>end node</i> posisi Ketiga.....	102
Tabel 4. 27 Hasil Pengujian <i>end node</i> posisi Keempat.....	103
Tabel 4. 28 Hasil Pengujian <i>end node</i> posisi Kelima.....	104
Tabel 4. 29 Hasil Analisis Pengujian Skenario Ketiga	105



UNIVERSITAS
Dinamika

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Program End Node 1	111
Lampiran 2 Program End Node 2	116
Lampiran 3 Program End Node 3	121
Lampiran 4 Datacake Skenario 1	126
Lampiran 5 Datacake Skenario 2	131
Lampiran 6 Datacake Skenario 3	135
Lampiran 7 Data Skenario 1	137
Lampiran 8 Data Skenario 2	139
Lampiran 9 Data Skenario 3	142
Lampiran 10 Hasil Plagiasi	144
Lampiran 11 Kartu Bimbingan	148



UNIVERSITAS
Dinamika

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pesisir pantai seringkali merupakan area yang sulit dijangkau oleh infrastruktur komunikasi jaringan seluler. Hal ini dapat disebabkan oleh faktor geografis, termasuk topografi yang sulit atau jarak yang jauh dari pusat perkotaan. Keterbatasan ini dapat menghambat akses komunikasi yang efektif di wilayah pesisir. Pesisir pantai sering menjadi area yang penting untuk dipantau, baik untuk tujuan pengelolaan lingkungan, keamanan, maupun kepentingan ekonomi seperti perikanan. Oleh karena itu, adopsi sistem komunikasi yang dapat mendukung pengiriman data dari perangkat sensor di pesisir pantai menjadi sangat relevan. Salah satunya adalah menerapkan jaringan komunikasi LoRa. Jaringan LoRa dapat dikombinasikan dengan teknologi komunikasi yang eksisting untuk menghasilkan jangkauan yang lebih luas (Puspitorini, Mahmudah, Wijayanti, & Adi, 2023).

LoRa merupakan komunikasi nirkabel dengan modulasi *Chirp Spread Spectrum (CSS)*. LoRa dapat dioperasikan secara *free license* pada pita sub-gigahertz, misalnya 915 MHz, 868 MHz, dan 433 MHz. LoRa dapat beroperasi pada frekuensi 2,4 GHz untuk kecepatan data yang lebih tinggi. Sedangkan pengembangan yang baru dari LoRa adalah LoRaWAN (Alliance, *what is LoRaWAN spesification*, 2024). LoRaWAN adalah protokol yang dibangun dari modulasi LoRa pada *layer Media Access Control (MAC)*. LoRaWAN pertama kali muncul pada tahun 2015, yang dikembangkan oleh LoRa Alliance. LoRaWAN dapat dihubungkan ke internet pada skala regional, nasional, atau global (The Things Network, 2024).

LoRaWAN dapat terhubung dengan perangkat sensor atau *actuator* melalui koneksi internet dan server. Platform *cloud server* yang terhubung dengan LoRaWAN adalah *TheThingsNetwork*, *wisDM*, *ChirpStack*, *Amazon Web Services (AWS)*, dan *Azure*. Penelitian yang menerapkan LoRaWAN untuk mengontrol kualitas air di daerah pedesaan dengan menggunakan server *The Things Network* (Waheb A.Jabbar, 2024). Hasil penelitan tersebut LoRaWAN membuat sistem

monitoring lebih efisien dan dapat diakses dengan menggunakan berbagai platform IoT.

Perangkat sensor mengirimkan data ke LoRaWAN secara real-time. Pada proposal ini LoRaWAN akan menerima data posisi dari perangkat *end node* yang terpasang sensor GPS. GPS dapat mengetahui koordinat lintang dan koordinat bujur dari suatu tempat atau titik dipermukaan bumi. Terdapat penelitian sebelumnya yang mengirimkan data GPS menggunakan transmisi LoRa (Harjowinoto, 2021). Pada penelitian tersebut menggunakan format GPS latitude dan longitude, menghasilkan rata-rata paket loss lebih dari 30% dimana nilai tersebut masih tergolong besar. Format koordinat dari GPS dapat dirubah menjadi fungsi jarak, fungsi jarak tersebut dapat menggunakan metode *haversine*. Metode *haversine* merupakan salah satu metode dalam pengukuran jarak antar dua sumbu titik koordinat dengan data acuan garis bujur dan garis lintang. Metode ini dianggap cukup akurat untuk penerapan pembacaan data sensor GPS (Setiadi, 2023). Pada Penelitian yang dilakukan oleh (Nugraha, 2023), membuktikan penggunaan metode *haversine* yang diterapkan dalam pencarian toko obat terdekat menggunakan aplikasi untuk menentukan jarak antara dua titik menghasilkan akurasi hingga 99,97%.

Pada penelitian yang dilakukan oleh (Harjowinoto, 2021), data GPS dikirimkan oleh transmisi antara dua *end node* LoRa. Sehingga seluruh data yang diterima oleh salah satu *node* kemudian melakukan proses pengolahan data dan tidak dikirimkan menuju server, dapat disimpulkan penelitian sebelumnya belum mendukung sistem LoRaWAN. Untuk menerapkan sistem LoRaWAN membutuhkan sebuah gateway yang dapat terhubung ke server, sehingga data dapat disimpan dan diolah oleh server. Tujuan dari Tugas Akhir ini adalah menerapkan LoRaWAN sebagai *gateway* yang menerima data posisi dan jarak dari dua perangkat *end node* LoRa yang terpasang sensor GPS. Dalam penentuan jarak dari *end node* terhadap *gateway* ditentukan oleh metode *haversine*. Hasil perhitungan jarak tersebut akan dibandingkan dengan perangkat komersial GPS Garmin. Selain itu, proposal ini juga menguji performa komunikasi *end node* dengan LoRaWAN di daerah pesisir Pantai.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dapat dirumuskan permasalahan dalam Tugas Akhir ini adalah:

1. Bagaimana menerapkan metode *haversine* pada perangkat *end node*?
2. Bagaimana menganalisis performa komunikasi perangkat *end node* LoRa terhadap *gateway* LoRaWAN di lingkungan pesisir Pantai?

1.3 Batasan Masalah

Dalam pembuatan Tugas Akhir ini, ruang lingkup penelitian hanya akan dibatasi pada:

1. Perangkat *end node* yang digunakan sebanyak tiga.
2. Konsumsi daya dari perangkat *end node* tidak dianalisis.
3. Mode aktivasi pada perangkat *end node* ke *gateway* menggunakan *Over-The-Air – Activation* (OTAA).

1.4 Tujuan

Berdasarkan uraian latar belakang dan rumusan masalah di atas, maka tujuan dari Tugas Akhir ini yaitu sebagai berikut:

1. Mengimplementasikan metode *haversine* dalam bahasa pemrograman pada *end node* LoRa yang terpasang sensor GPS agar menghasilkan data jarak antara dua titik dengan presisi yang tinggi.
2. Menguji performa komunikasi *end node* LoRa dengan LoRAWAN dengan parameter analisis RSSI, SNR, *Spreading Factor*, dan *Airtime*.

1.5 Manfaat

Adapun manfaat penyusunan Tugas Akhir ini adalah menerapkan komunikasi LPWAN untuk pemantauan posisi wisatawan khususnya di daerah pesisir pantai dalam rangka peningkatan keamanan di sektor *tourism*.

BAB II

LANDASAN TEORI

2.1 Infrastruktur Komunikasi di daerah Pesisir Pantai

Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) Wilayah Jatim mengatakan bahwa, beberapa wilayah pesisir Pantai di Jawa Timur masih kurang pendirian akses internet (KOMINFO, 2024).

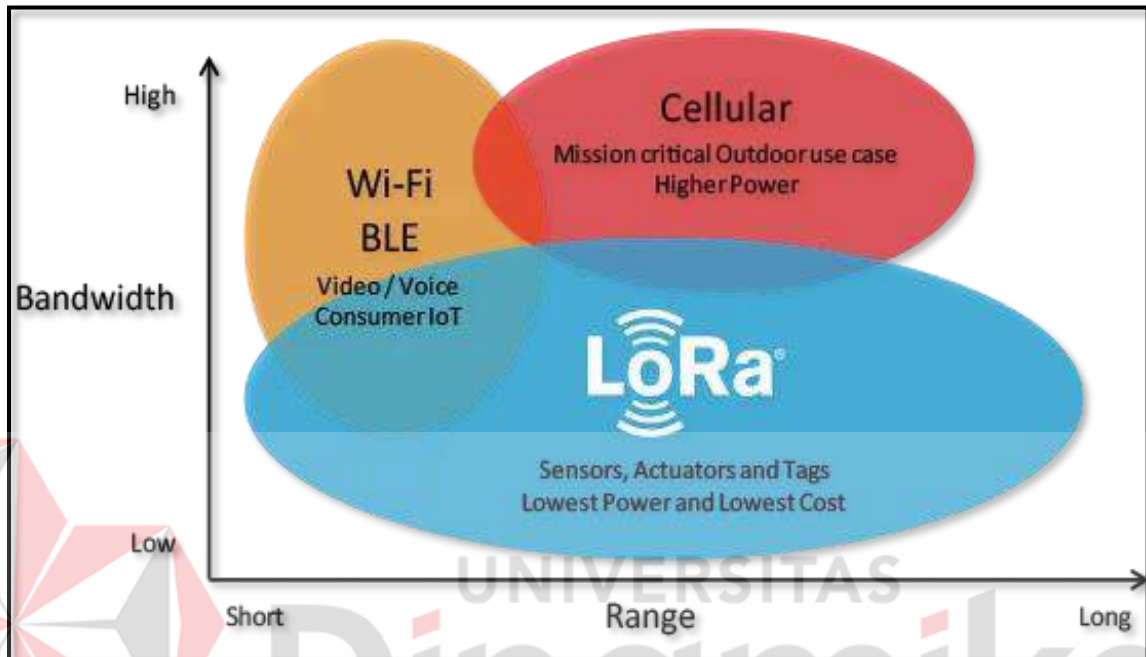
Kekurangan akses internet menyebabkan perkembangan informasi mengalami kesulitan. Akibat kurangnya akses internet di daerah pesisir pantai menyebabkan kurangnya penggunaan fasilitas umum yang menggunakan jaringan internet, sehingga APJII menilai sumber daya manusia (SDM) di daerah pesisir pantai belum bisa menggunakan akses internet dengan maksimal.

Saat ini, akses internet masih memanfaatkan adanya koneksi teknologi seluler seperti 4G atau 5G, dimana secara penerapan teknologi tersebut membutuhkan biaya yang cukup tinggi. Sehingga, sangat dimungkinkan digabungkan dengan komunikasi yang penerapannya lebih murah dan mudah (Puspitorini, Mahmudah, Wijayanti, & Adi, 2023). Salah satu komunikasi yang penerapannya mudah adalah komunikasi LoRa.

2.2 LoRa

Long Range (LoRa) adalah media komunikasi nirkabel berdaya rendah yang menggunakan modulasi *Chirp Spread Spectrum* (CSS). LoRa pertama kali diperkenalkan oleh Semtech pada tahun 2013. LoRa memiliki daya pancar yang mampu mencakup 3-7 km tergantung pada kondisi lingkungan dan berbagai fitur sehingga cocok untuk sensor dan *actuator* yang beroperasi pada daya rendah dan komersialisasi dengan biaya yang murah (Fahmida, 2023).

Gambar 2.1 menunjukkan beberapa media komunikasi nirkabel dan perkiraan jangkauan dari segi transmisi dan *bandwidth*. LoRa ideal untuk mengirimkan sejumlah data kecil dengan *bit rate* yang rendah. Data dapat ditransmisikan dari jarak yang lebih jauh dibandingkan media komunikasi atau teknologi seperti, WiFi, Bluetooth, atau ZigBee (Thethingsnetwork, 2024).



Gambar 2. 1 Jangkauan komunikasi nirkabel
(Sumber: Alliance, THE THINGS NETWORK, 2024)

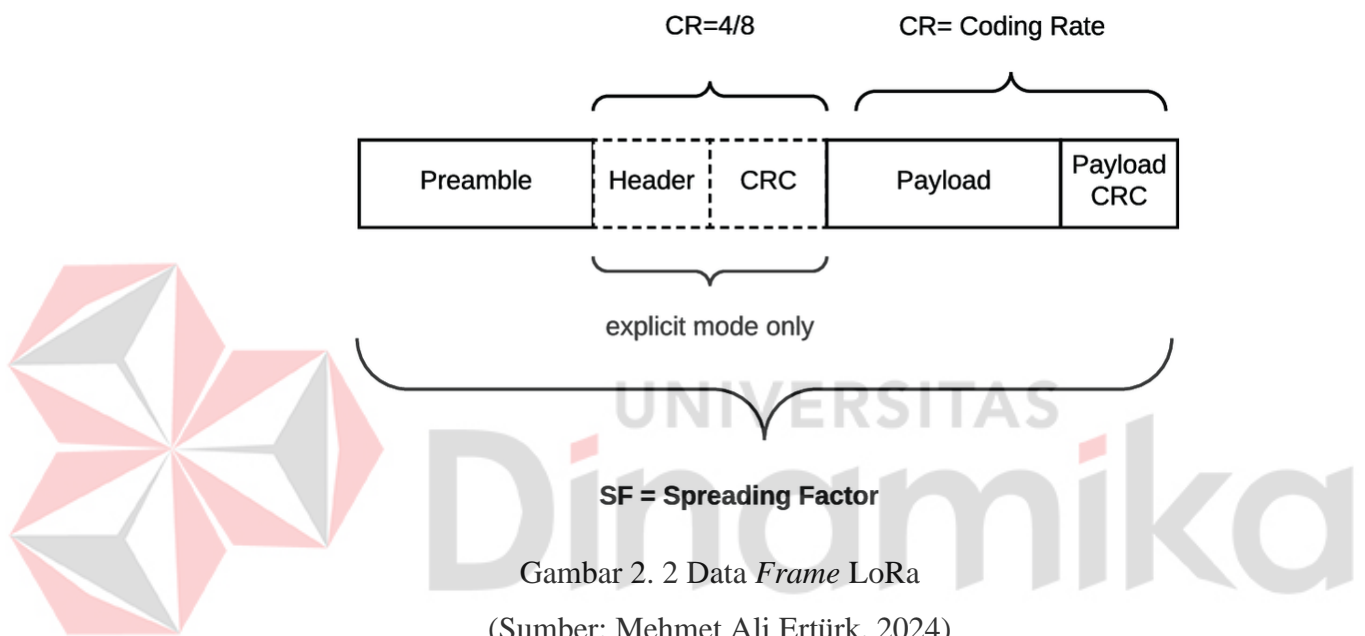
Berdasarkan Gambar 2.1, komunikasi *wireless* dibagi menjadi tiga kategori, yaitu komunikasi *short-range*, komunikasi seluler, dan komunikasi *Low Power Area Network* (LPWAN). Perbandingan ketiga kategori tersebut ditunjukkan pada Tabel 2.1.

Tabel 2. 1 Perbandingan komunikasi nirkabel

Teknologi Wireless	Wireless Communication	Range	Tx Power
Bluetooth	<i>short-range</i>	~ 10 meter	~ 2,5 dB
WIFI	<i>short-range</i>	~ 50 meter	~ 80 dB
3G/4G	seluler	~ 5000	~ 500 dB
LoRa	LPWAN	1. 2k-5k daerah urban	~ 20 dB

-
2. 5k-1,5k daerah rural
 3. >15k posisi *Line of Sight* (LoS)
-

Pada Tugas Akhir ini akan memakai komunikasi LoRa dengan settingan frekuensi 915/923 MHz, agar LoRa dapat bekerja dengan baik dan sesuai dengan peraturan pemerintah.



Gambar 2. 2 Data *Frame* LoRa

(Sumber: Mehmet Ali Ertürk, 2024)

Gambar 2.2 menunjukkan struktur data *frame* pada protokol transmisi LoRa. Berikut ini adalah penjelasan mengenai bagian *frame* dari struktur data LoRa:

1. *Preamble* merupakan bagian awal dari *frame* yang digunakan untuk sinkronisasi antara pengirim dan penerima. *Preamble* terdiri dari urutan bit yang memungkinkan penerima untuk mendeteksi sinyal dan mempersiapkan diri untuk menerima data.
2. *Header* berisi informasi penting tentang *frame*, seperti panjang *payload* dan beberapa parameter konfigurasi lainnya. Pada mode eksplisit, *header* ini selalu ada dan dilengkapi dengan mekanisme koreksi kesalahan (*Coding Rate*).

3. CRC (*Cyclic Redundancy Check*) adalah kode yang digunakan untuk mendeteksi kesalahan dalam header. CRC ini membantu memastikan bahwa data yang diterima tidak rusak selama transmisi.
4. *Payload* adalah bagian dari *frame* yang mengandung data sebenarnya yang dikirim. Data ini bisa berupa informasi sensor, pesan, atau data lainnya yang diperlukan oleh aplikasi.
5. *Payload* CRC mirip dengan CRC pada *header*, *Payload* CRC digunakan untuk memverifikasi integritas data pada *payload*. Ini memastikan bahwa data *payload* tidak mengalami kerusakan selama transmisi.

2.3 LoRa Gateway

LoRa gateway adalah perangkat keras yang digunakan dalam jaringan LoRaWAN untuk menghubungkan perangkat *end node* yang menggunakan teknologi LoRa dengan server jaringan LoRaWAN (LNS). LoRa gateway bertindak sebagai pintu gerbang antara perangkat-perangkat LoRa yang tersebar luas di lapangan dengan infrastruktur jaringan yang terhubung ke internet. Fungsi utama LoRa gateway adalah:

1. Menerima dan mendekode data yang dikirim oleh perangkat LoRa yang beroperasi dalam jangkauan radio.
2. Mengirim data yang diterima dari perangkat *end node* LoRa ke server atau *platform cloud* melalui koneksi internet.
3. Memproses data dari server atau *platform cloud* untuk dikirimkan kembali ke perangkat *end node* LoRa yang sesuai.
4. Memantau dan mengelola status jaringan LoRaWAN, termasuk koordinasi pengiriman dan penerimaan data antara perangkat dan server.

LoRa gateway umumnya dilengkapi dengan antarmuka jaringan seperti *Ethernet* atau Wi-Fi untuk terhubung ke internet, serta antarmuka radio LoRa untuk berkomunikasi dengan perangkat LoRa *end node*. Ini memungkinkan LoRa gateway untuk beroperasi sebagai penghubung kritis dalam infrastruktur jaringan LoRaWAN yang mencakup area yang luas (Leibbrandt, 2024).

2.4 *The Things Network* (TTN)

The Things Network (TTN) adalah sebuah komunitas global yang membangun dan mengoperasikan jaringan LoRaWAN *open source*. Tujuan utama TTN adalah menyediakan akses terbuka dan gratis ke jaringan LoRaWAN untuk mendukung pengembangan solusi IoT (*Internet of Things*). *The Things Network* bertujuan untuk mengurangi hambatan akses dan mendorong inovasi dalam pengembangan solusi IoT dengan menyediakan infrastruktur jaringan yang terbuka, gratis, dan mudah diakses bagi siapa pun di seluruh dunia (Alliance, THE THINGS NETWORK, 2024).



Gambar 2. 3 *The Things Network*
(Sumber: Alliance, THE THINGS NETWORK, 2024)

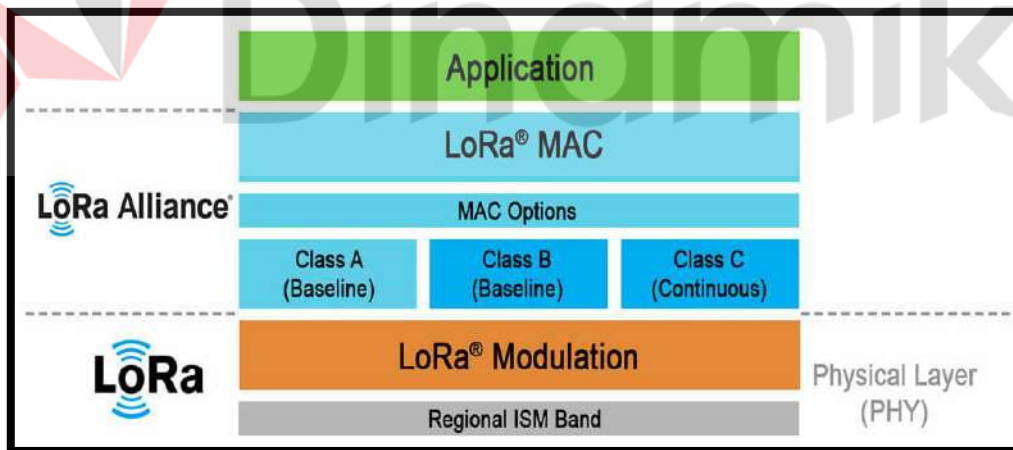
Ada beberapa alasan Tugas Akhir ini memilih platform *The Things Network* (TTN) dibandingkan dengan platform Antares dan lainnya:

- **Komunitas yang Luas:** TTN memiliki komunitas yang luas dan dapat dengan mudah menemukan dukungan dan sumber daya dari pengguna lain yang memiliki pengalaman dengan platform ini.
- **Model Bisnis dan Keberlanjutan:** TTN didasarkan pada model bisnis *open-source* dan berkelanjutan sehingga platform ini akan terus berkembang dan didukung oleh komunitas yang ada.

- **Dukungan Keamanan:** TTN memiliki dukungan untuk autentikasi *Over The Air Activation (OTAA)* dan *Activation By Personalization (ABP)* yang membuat mekanisme keamanan menjadi lebih yang tinggi.
- **Skalabilitas dan Ketersediaan:** TTN telah terbukti dapat menangani jumlah perangkat yang besar dan tersedia di banyak wilayah di seluruh dunia.
- **Biaya:** TTN sering kali menawarkan model gratis atau berbayar dengan opsi tambahan.

2.5 LoRaWAN

LoRaWAN merupakan salah satu telekomunikasi dari IoT yang memiliki standar protokol tersendiri yaitu, *Media Access Control (MAC)* yang dibangun diatas modulasi LoRa, agar perangkat *end node* dan *LoRa gateway* dapat melakukan komunikasi. Berdasarkan OSI layer, LoRaWAN menghubungkan komunikasi mencakup layer *physical* hingga layer *network*.



Gambar 2. 4 *Physical and Communication layers of a LoRaWAN Network*

(Sumber: Przerwa Urlopowa, 2024)

Pada *physical layer* LoRaWAN menggunakan modulasi LoRa yang bekerja pada pita frekuensi ISM dan menggunakan teknik *spread spectrum* untuk menyebarkan sinyal ke berbagai frekuensi, mengurangi interferensi. Pada layer MAC, LoRaWAN menerapkan beberapa opsi kelas seperti Class A, B, dan C yang mengatur waktu transmisi dan penerimaan data, dengan Class B dan C

menggunakan teknik *Time-Division Multiple Access* (TDMA) untuk memberikan slot waktu tertentu pada perangkat. Selain itu, LoRaWAN menggunakan *Adaptive Data Rate* (ADR) untuk menyesuaikan laju data perangkat berdasarkan kondisi saluran, serta versi yang dimodifikasi dari protokol Aloha, di mana perangkat mengirim data secara acak dan mengirim ulang jika terjadi tabrakan, memastikan komunikasi yang efisien dan minimal risiko tabrakan dalam jaringan.

LoRaWAN beroperasi dengan menggunakan jalur frekuensi Industri, Ilmiah, dan medis (ISM) yang *free lincense*. List Frekuensi *plan* sesuai nama *regional* pada umumnya ditunjukkan pada Tabel 2.2.

Tabel 2. 2 *List Frekuensi Plan*

<i>Plan ID</i>	<i>Frequency Plan</i>	<i>Common Name</i>
1	EU863-870	EU868
2	US902-928	US915
3	CN779-787	CN779
4	EU433	EU433
5	AU915-928	AU915
6	CN470-510	CN470
9	AS923-3	AS923-3
10	KR920-923	KR920
11	IN865-867	IN865
12	RU864-870	RU864
13	AS923-4	AS923-4

Spesifikasi LoRaWAN didefinisikan menjadi tiga jenis yaitu, kelas A, kelas B, dan kelas C. LoRaWAN secara umum wajib menerapkan spesifikasi dari kelas A, sedangkan kelas B dan C merupakan perluasan dari spesifikasi perangkat kelas A. Spesifikasi kelas perangkat LoRaWAN ditunjukkan pada Tabel 2.3.

Tabel 2. 3 Spesifikasi Kelas Perangkat LoRaWAN

Fitur	Kelas A	Kelas B	Kelas C
<i>Message Delivery</i>	<i>Downlink</i>	<i>Downlink</i>	<i>Downlink</i>
<i>Receiving</i>	<i>Up+Down</i>	<i>Up+Down</i>	<i>Up+Down</i>
<i>Synchronization</i>	Tidak	Periodik	Tidak
<i>Power Consumption</i>	Rendah	Sedang	Tinggi
<i>Priority</i>	Rendah	Sedang	Tinggi

Pada Tugas Akhir ini, terdapat beberapa parameter analisis koneksi *end node* ke LoRAWAN, yaitu:

1. *Received Signal Strength* indicator (RSSI), merupakan pengukuran untuk menentukan sinyal yang diterima cukup kuat untuk memperoleh koneksi nirkabel yang baik dari pemancar. RSSI diukur dalam dBm dan nilainya dalam bentuk negatif, jika semakin dekat nilai RSSI ke angka nol, maka sinyal yang diterima semakin kuat. Persamaan untuk perhitungan rata-rata RSSI dilihat pada persamaan (1) (Affandi, 2020).

$$RSSI(dBm) = \frac{\text{Total Nilai RSSI yang diterima}}{\text{Jumlah Sampel RSSI yang dikirim}} \quad (1)$$

2. Signal to Ratio (SNR), merupakan parameter agar menentukan kualitas sinyal yang terganggu noise. Persamaan untuk perhitungan rata-rata *Signal to Ratio* dilihat pada persamaan (2)-(3).

$$SNR(dB) = 10 \log_{10} \left(\frac{P_{\text{signal}}}{P_{\text{noise}}} \right) \quad (2)$$

$$P = \text{daya}(dB) \quad (3)$$

3. *Code Rate* (CR) adalah salah satu parameter penentu untuk meningkatkan nilai *Forward Error Correction* (FEC). CR dapat diimplementasi dengan merubah proses *encode*, nilai *encode* dapat dirubah dengan rentang 4bit sampai 8 bit. Semakin besar nilai *encode*, maka sinyal LoRa menjadi lebih tahan terhadap interferensi. Namun semakin besar nilai CR akan mempengaruhi *airtime* (setiawan, 2020).

4. *Tx Power* merupakan besar daya yang digunakan antenna LoRa saat pengiriman dan satuan pengukuran *Tx Power* adalah *decible* meter (dBm). Semakin rendah *Tx Power* maka lebih menghemat penggunaan baterai, namun jarak sinyal antenna LoRa yang dipancarkan akan semakin pendek, sebaliknya jika semakin besar *Tx Power*, jarak sinyal dari antenna LoRa yang dipancarkan akan semakin jauh (Thethingsnetwork, 2024).
5. *Data Rate* merupakan parameter yang merujuk pada kecepatan transfer data antara perangkat *end node* LoRa dan LoRa *gateway*. *Data rate* dapat dipengaruhi oleh beberapa faktor, termasuk konfigurasi perangkat, jarak antara perangkat *end node* LoRa ke LoRa *gateway*, dan kondisi lingkungan. Lebih tinggi *data rate* maka, semakin cepat perangkat *end node* dapat mentransfer data, tetapi biasanya mengakibatkan jangkauannya menjadi lebih pendek.
6. *Frame Counter* (FCNT) merupakan nilai *frame* yang digunakan dalam proses pengiriman data menggunakan autentikasi OTAA maupun ABP. Nilai FCNT pada autentikasi ABP akan sama pada setiap pengiriman datanya, sedangkan autentikasi OTAA menggunakan nilai FCNT yang berbeda dalam setiap pengiriman datanya sehingga *frame* yang digunakan akan bertambah secara dinamis sehingga pengiriman data akan menjadi lebih aman.
7. *Airtime* merupakan, parameter yang digunakan untuk mengukur selisih paket data yang diterima dan data yang dikirim, Dimana sinyal dikirim dari pengirim dalam waktu tertentu sebelum sinyal diterima oleh penerima. Perhitungan *airtime* dilihat pada persamaan (4).
$$\text{Time On Air}(ms) = \text{Waktu terima} - \text{Waktu kirim} \quad (4)$$
8. *Frameport* merupakan bagian dari pesan LoRaWAN yang digunakan untuk menentukan penggunaan data kustom. FPort adalah bilangan bulat antara 1 sampai 223, yang berarti ada 223 slot yang tersedia untuk aplikasi pengguna. FPort berfungsi agar server aplikasi dapat membedakan antara jenis data yang dikirim oleh perangkat *end node* dan menangani data tersebut sesuai dengan kebutuhan server aplikasi.

2.6 Mode Aktivasi *End Node* ke TNN

Mode aktivasi dari perangkat *end node* ke server TNN, perlu dilakukan registrasi ke server dengan dua mode yaitu *Activation By Personalization* (ABP) dan *Over The Air Activation* (OTAA).

2.6.1 *Activation By Personalization* (ABP)

ABP adalah metode alternatif untuk mengaktifkan perangkat secara manual untuk terhubung jaringan LoRaWAN tanpa melalui proses Aktivasi *Over-The-Air* (OTAA). Sebagai gantinya, perangkat diberikan parameter keamanan yang diperlukan termasuk DevAddr, NwkSKey, dan AppSKey pada perangkat end node.



Gambar 2. 5 *Activation By Personalization*

(Sumber: Alliance, *THE THINGS NETWORK*, 2024)

Berikut adalah penjelasan singkat tentang komponen-komponen kunci yang terlibat dalam ABP:

1. DevAddr (*Device Address*)

- DevAddr berfungsi sebagai pengenal unik yang ditugaskan ke perangkat dalam jaringan LoRaWAN.
- Digunakan dalam *header* pesan LoRaWAN untuk menunjukkan perangkat tujuan atau sumber.

2. NwkSKey (*Network Session Key*)

- NwkSKey adalah kunci sesi yang digunakan antara perangkat dan server jaringan LoRaWAN untuk mengamankan komunikasi di lapisan jaringan.

- Digunakan untuk mengenkripsi dan mendekripsi pesan yang dikirim antara perangkat dan server jaringan.

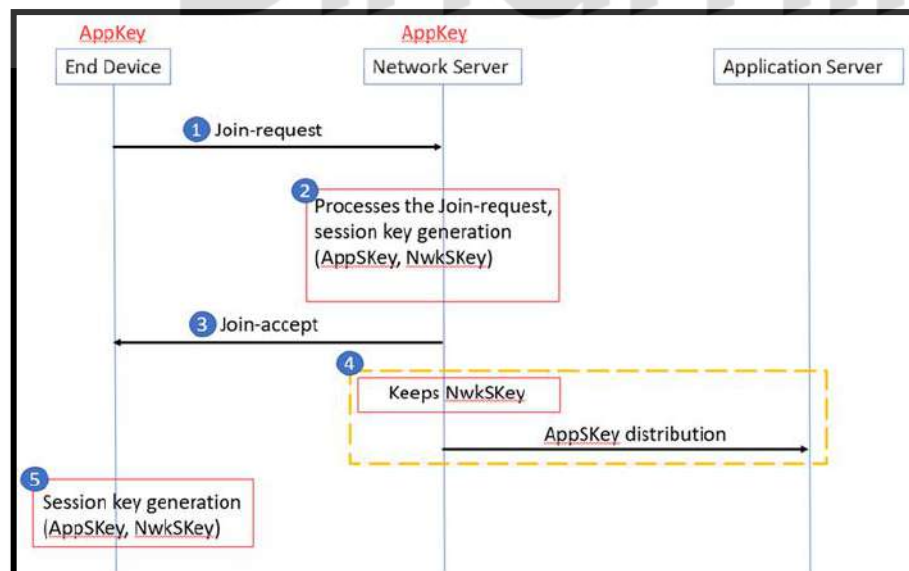
3. AppSKey (*Application Session Key*)

- AppSKey adalah kunci sesi yang digunakan antara perangkat dan server aplikasi LoRaWAN untuk mengamankan komunikasi di lapisan aplikasi.
- Digunakan untuk mengenkripsi dan mendekripsi *payload* data yang dikirim antara perangkat dan server aplikasi.

ABP memungkinkan perangkat untuk terhubung dengan jaringan LoRaWAN. Tetapi konfigurasi ABP memberikan perangkat *end node* parameter keamanan secara manual dan memunculkan resiko keamanan tertentu, seperti potensi untuk paparan kunci selama perangkat *end node* aktif. Selain itu, ABP kurang memiliki beberapa manfaat OTAA, seperti pertukaran kunci dinamis dan otentikasi perangkat selama proses aktivasi.

2.6.2 *Over The Air Activation (OTAA)*

OTAA adalah proses dimana sebuah perangkat harus memiliki tiga elemen kunci DevEUI, AppEUI, dan AppKey untuk dapat terhubung dengan jaringan LoRaWAN.



Gambar 2. 6 *Over The Air Activation*

(Sumber: Alliance, *THE THINGS NETWORK*, 2024)

- DevEUI: identifikasi unik global 64-bit dari perangkat.
- AppEUI: identifikasi unik 64-bit yang dimiliki oleh pemilik aplikasi yang ingin perangkat terhubung.
- AppKey: adalah kunci 128-bit yang digunakan untuk menandatangani permintaan bergabung awal, termasuk *DevEUI*, *AppEUI*, dan *DevNonce* (sebuah angka acak dua byte yang dihasilkan).

Setelah permintaan bergabung awal dikirimkan oleh perangkat, server jaringan akan memeriksa kevalidan permintaan tersebut. Jika valid, server akan merespons dengan mengirimkan beberapa informasi kunci yang diperlukan oleh perangkat untuk bergabung:

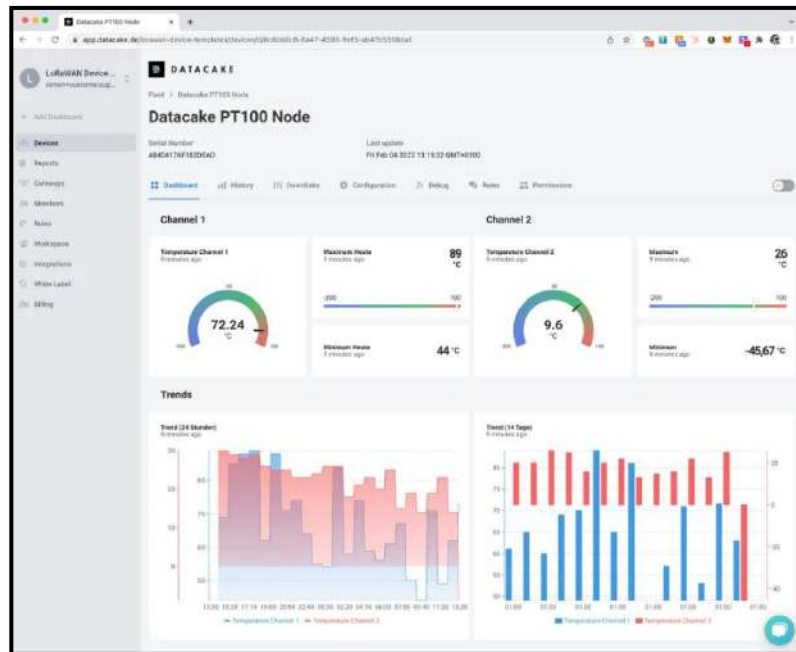
- *AppNonce*: Sebuah angka acak 128-bit yang digunakan dalam proses enkripsi.
- *AppSKey*: Kunci sesi aplikasi 128-bit yang digunakan untuk enkripsi data aplikasi.
- *NwkSKey*: Kunci sesi jaringan 128-bit yang digunakan untuk enkripsi data jaringan.
- *DevAddr*: Alamat unik perangkat dalam jaringan LoRaWAN.
- *RxDelay*: Penundaan penerimaan yang digunakan oleh perangkat untuk menentukan kapan membuka jendela penerimaan.
- *CFList*: Daftar frekuensi (*channel*) yang harus digunakan oleh perangkat.

Over The Air Activation (OTAA) memungkinkan perangkat untuk terhubung dengan jaringan LoRaWAN secara dinamis dan aman, dengan menggunakan kunci yang dihasilkan secara dinamis dan unik untuk setiap sesi koneksi.

2.6.3 Perbedaan ABP dan OTAA

Konfigurasi ABP memprogram parameter otentikasi ke dalam perangkat *end node* tanpa pertukaran pesan otentikasi saat aktivasi. Lebih sederhana tetapi kurang aman. Sedangkan konfigurasi menggunakan OTAA melibatkan pertukaran pesan otentikasi dengan server jaringan LoRaWAN, menghasilkan kunci otentikasi secara dinamis. Lebih aman dan fleksibel (Blenn & Kuipers, 2017).

2.7 Datacake



Gambar 2. 7 Datacake

(Sumber: Datacake, 2024)

Datacake adalah platform IoT (*internet of things*) yang dirancang untuk membantu pengguna mengelola perangkat IoT dan data yang dihasilkan oleh perangkat tersebut. Platform ini menyediakan fitur yang memudahkan proses pengumpulan, penyimpanan, pemrosesan, analisis, dan visualisasi data IoT.

2.8 Arduino Due



Gambar 2. 8 Arduino Due Board

(Sumber: Arduino Indonesia ID, 2024)

Arduino Due adalah papan mikrokontroler yang berbasis pada mikrokontroler Atmel SAM3X8E ARM Cortex-M3. Arduino Due dirancang untuk

aplikasi yang membutuhkan lebih banyak daya pemrosesan dan kemampuan I/O yang lebih baik. Sehingga Arduino Due lebih efisien dalam pengolahan data GPS yang kompleks. Spesifikasi perangkat Arduino Due ditunjukkan pada tabel 2.4 (Arduino, 2024).

Tabel 2. 4 Spesifikasi Arduino Due

Spesifikasi	Detail
Mikrokontroler	Atmel SAM3X8E ARM Cortex-M3
Tegangan Operasional	3.3V
Tegangan Input	7-12V
Pin Digital I/O	54 (12 pin dapat digunakan sebagai PWM output)
Pin Input Analog	12
Pin Output Analog	2
Pin PWM	12
UART	4
Memori Flash	512 KB (dua bagian dari 256 KB)
SRAM	96 KB (64 KB + 32 KB)
Kecepatan Clock	84 MHz
Dimensi	101.52 mm x 53.3 mm
Berat	36 gram

2.9 Global Positioning System (GPS)



Gambar 2. 9 GPS NEO-6M

(Sumber: Arduino, 2024)

Global Positioning System (GPS) adalah sistem navigasi satelit yang memiliki jaringan satelit di orbit Bumi untuk menentukan posisi. Sistem ini awalnya dikembangkan dan dioperasikan oleh militer Amerika Serikat tetapi sekarang telah menjadi sistem navigasi global yang tersedia untuk pengguna sipil di seluruh dunia. Salah satu seri modul GPS yang sering digunakan umumnya adalah GPS NEO-6M dari u-blox. GPS NEO-6M memiliki prinsip kerja GPS pada umumnya. Selain itu, modul GPS NEO-6M juga dilengkapi dengan fitur-fitur tambahan seperti kemampuan integrasi dengan sistem mikrokontroler melalui komunikasi serial UART (*Time-To-First-Fix*), kemampuan untuk memperkirakan waktu awal penerimaan sinyal UART serta teknologi untuk mengurangi gangguan dan efek *multipath* di lingkungan yang sulit. Untuk spesifikasi perangkat GPS NEO-6M ditunjukkan pada Tabel 2.5 (AG, 2024).

Tabel 2. 5 Spesifikasi GPS NEO-6M

Spesifikasi	Detail
Chipset GPS	u-blox NEO-6M
Tegangan Operasional	2.7V - 3.6V
Tegangan <i>Input</i>	3.3V
Kecepatan Baud <i>Default</i>	9600 bps
Antena	Antena eksternal aktif atau pasif

Jumlah Kanal	50 kanal
Pin Waktu untuk Perbaikan	Cold Start: 27 detik dan Warm Start: 1 detik
Posisi (TTFF)	
Akurasi Posisi	2.5 meter CEP
Kecepatan Maksimum	515 m/s
Ketinggian Maksimum	50,000 meter
Suhu Operasional	-40°C hingga 85°C
Pinout	VCC, GND, TX, RX, PPS
Dimensi	16 mm x 12.2 mm x 2.4 mm

2.10 Algoritma *Haversine*

Menurut (Hasanah & Suharso, 2023) Algoritma *haversine* adalah metode perhitungan yang digunakan untuk mengestimasi jarak antara dua titik pada permukaan bola, seperti Bumi. Dalam konteks sistem navigasi, algoritma ini sangat penting karena dapat memberikan hasil jarak yang cukup akurat antara dua titik, dengan memperhitungkan radius Bumi. Algoritma ini memanfaatkan data titik lokasi yang dinyatakan dalam garis bujur (*longitude*) dan garis lintang (*latitude*), yang umumnya diperoleh dari perangkat GPS. Dengan menggunakan data koordinat ini, algoritma *haversine* dapat menghitung jarak antara dua titik dengan mempertimbangkan bentuk melengkung dari permukaan Bumi. Algoritma *haversine* dilihat pada persamaan (5)-(7).

$$a1 = \sin^2\left(\frac{\phi2 - \phi1}{2}\right) + \cos(\phi1) * \cos(\phi2) * \sin^2\left(\frac{\lambda2 - \lambda1}{2}\right) \quad (5)$$

$$a2 = 2 * \operatorname{atan2}(\sqrt{a1}, \sqrt{1 - a1}) \quad (6)$$

$$d(m) = a2 * R \quad (7)$$

Keterangan:

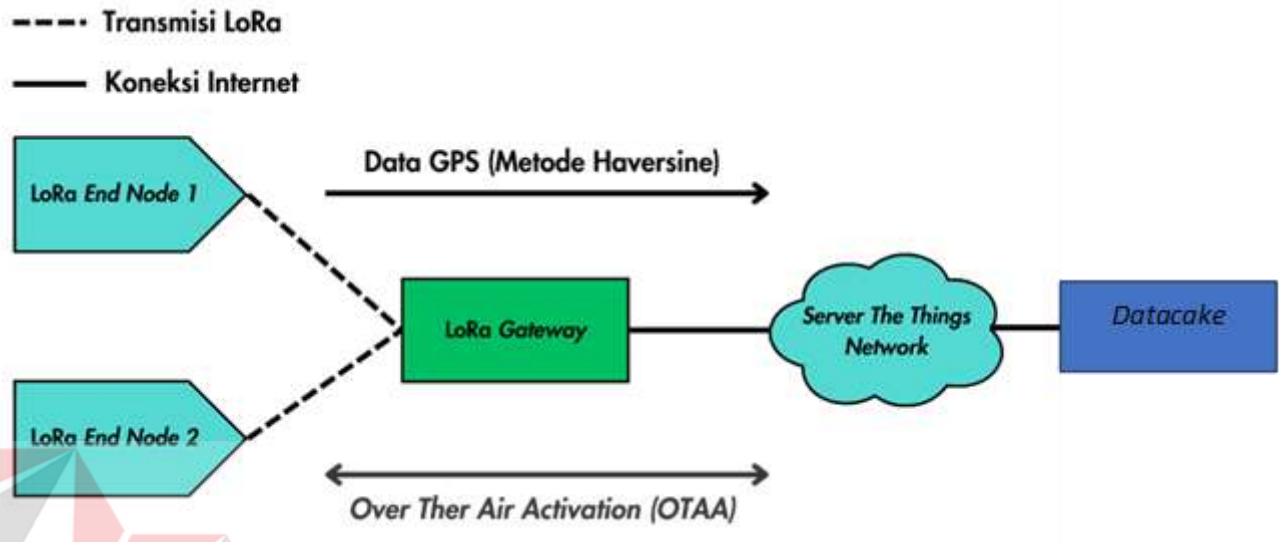
R = Jari-jari bumi (637000m) $\phi1$ = *Latitude* 1 (derajat)

d = Jarak (m) $\phi2$ = *Latitude* 2 (derajat)

$\lambda1$ = *Longitude* 1 (derajat) $\lambda2$ = *Longitude* 2 (derajat)

BAB III METODOLOGI PENELITIAN

3.1 Model Perancangan

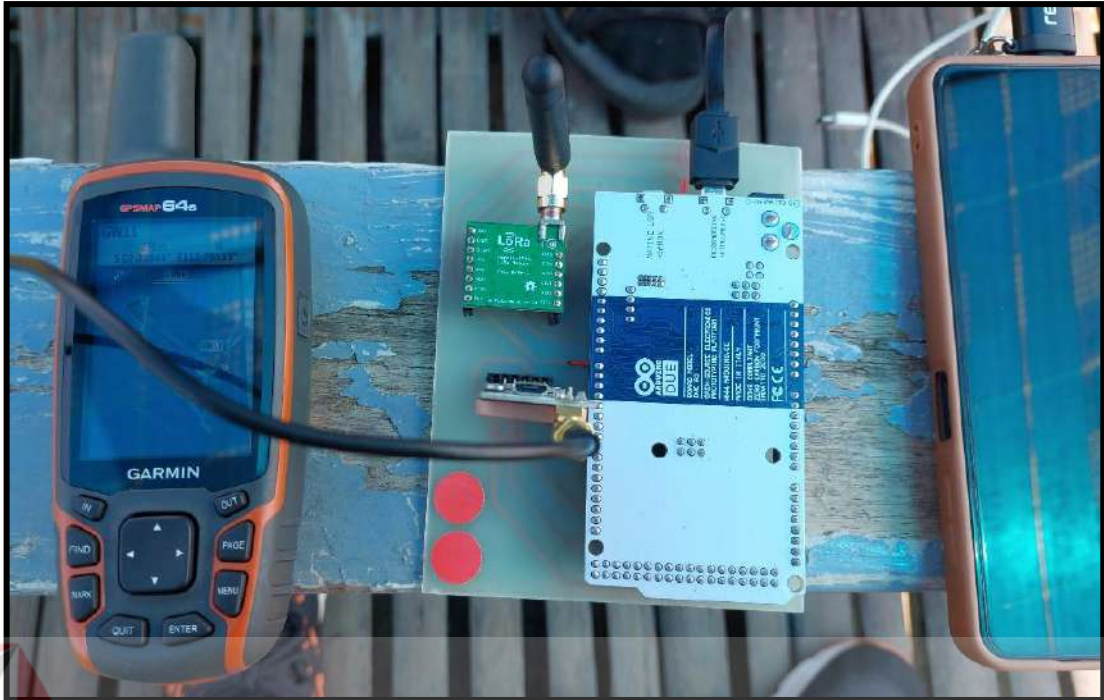


Gambar 3. 1 Model Perancangan Sistem

Gambar 3.1 menunjukkan model perancangan keseluruhan sistem yang akan dibuat pada Tugas Akhir ini. Tujuan utama dari penelitian ini adalah mengirimkan data dari sensor GPS yang telah diolah menggunakan algoritma *haversine* menuju *gateway* melalui media transmisi LoRa. Setelah menerima data GPS, *gateway* meneruskan data yang diterima menuju server *The Things Network* (TTN), yang kemudian diintegrasikan dengan *Datacake* agar menampilkan visualisasi data dengan yang rapih dan baik. Seluruh proses pengiriman akan menggunakan *OTAA device activation* mulai dari *end node* sampai *server* TTN, sehingga data yang dikirim dapat lebih terjamin keamanannya.

Terdapat beberapa bagian dari topologi yang telah ditentukan diantaranya adalah *LoRa end node*, *LoRa gateway*, dan server *The Things Network* (TTN), berikut penjelasan dari setiap bagian dari sistem yang dibuat.

3.1.1 LoRa End Node

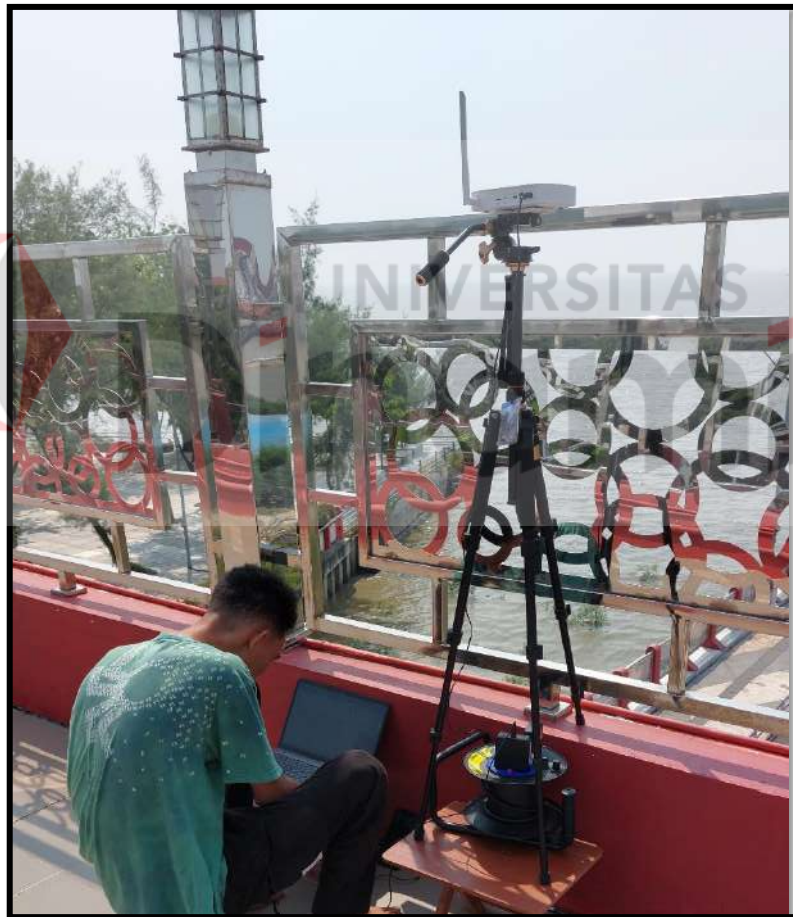


Gambar 3. 2 Perbandingan GPS Garmin dan Metode *Haversine*

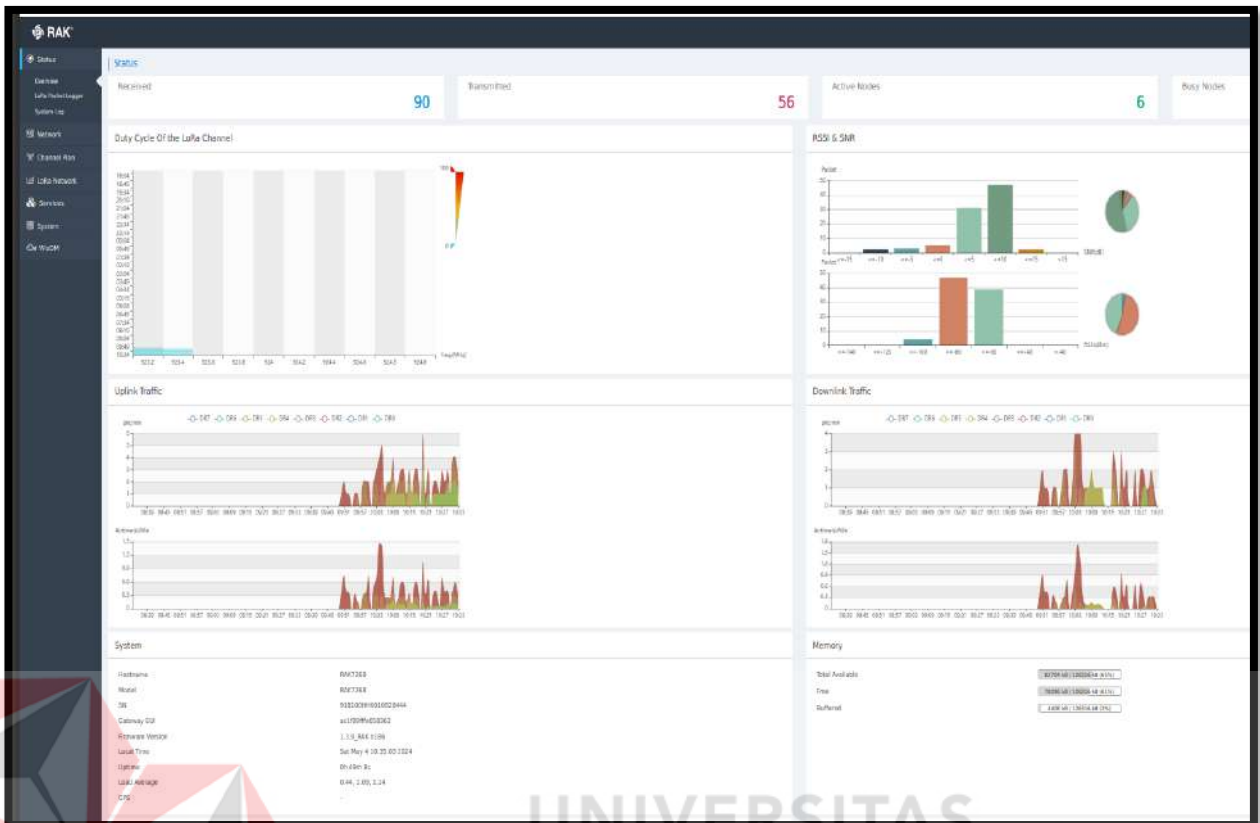
LoRa *end node* pada Tugas Akhir ini berfungsi untuk mengirimkan data GPS yang telah diolah menggunakan metode *haversine*. Metode *haversine* bertujuan untuk membantu menentukan jarak antar dua titik, yang pada Tugas Akhir ini adalah jarak LoRa *end node* pada LoRa *gateway*. LoRa *end node* mengirimkan data jarak dalam satuan meter menuju LoRa *gateway*, kemudian menuju server *The Things Network* yang kemudian diintegrasikan dengan *Datacake*. Hasil data jarak dari metode *haversine* dibandingkan dengan GPS Garmin kemudian dihitung selisih *error*. Rekap data dari end node tersimpan pada serial terminal yang terpasang pada *handphone*.

3.1.2 Gateway

Gateway berfungsi sebagai jembatan antara komunikasi transmisi LoRa dari *end node* dan juga komunikasi pada server melalui jaringan internet. LoRa *gateway* merupakan komponen penting pada Tugas Akhir ini, dimana LoRa *gateway* adalah jalur utama dari data yang dikirimkan oleh seluruh LoRa *end node*. LoRa *gateway* dapat menerima lebih dari satu LoRa *end node* secara bersamaan, dimana pada Tugas Akhir ini akan menerima data GPS dari beberapa LoRa *end node* yang berbeda secara bersamaan.



Gambar 3. 3 *Gateway*



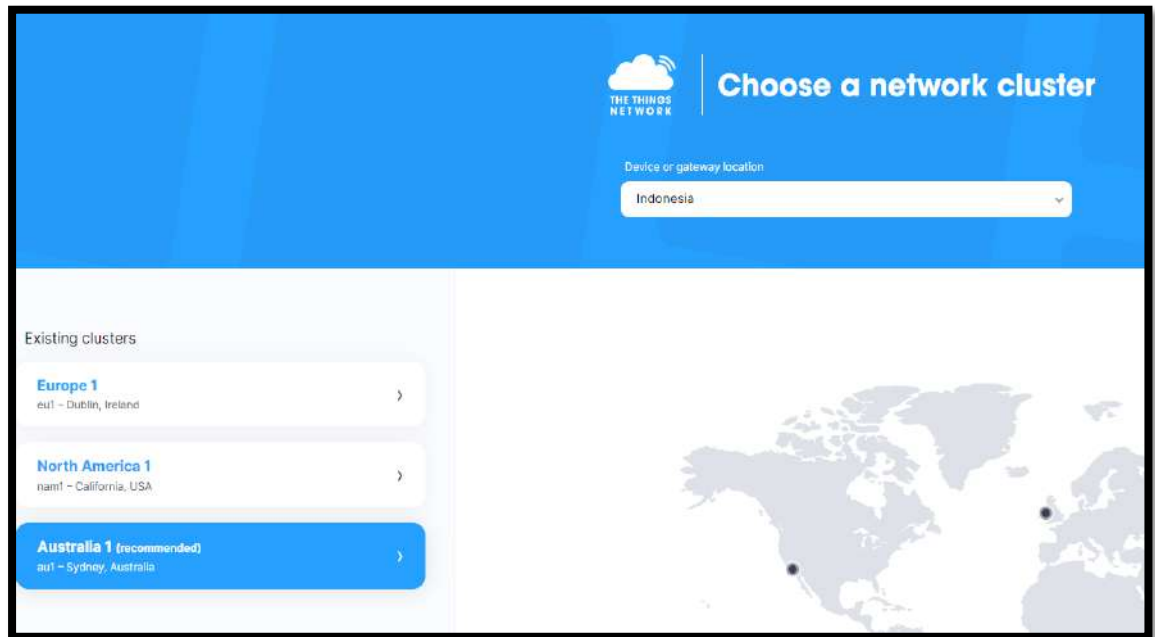
Gambar 3. 4 Dashboard Gateway

Data lokasi GPS akan melewati *gateway* terlebih dahulu lalu diteruskan menuju server TTN. *Gateway* harus terhubung pada internet untuk bisa mengakses server TTN, pada Tugas Akhir ini *gateway* dihubungkan ke internet melalui jaringan internet. Selain itu, akan dianalisis performa komunikasi *end node* ke *gateway* dengan parameter RSSI, SNR, *Spreading Factor*, dan *Airtime*.

3.1.3 Server The Things Network

Server *The Things Network* merupakan bagian yang menampung dan meneruskan data GPS yang dikirimkan oleh LoRa *end node* melalui LoRa *gateway* ke *Datacake*. Server TTN menjadi bagian utama dari proses autentikasi menggunakan *Over The Air Activation (OTAA)*, dimana server TTN sudah mendukung autentikasi OTAA dengan mendaftarkan beberapa *key* yang sama pada LoRa *end node* dan juga server TTN. Konfigurasi (OTAA) yang perlu dilakukan.

3.1.3.1 Pemilihan *Cluster*

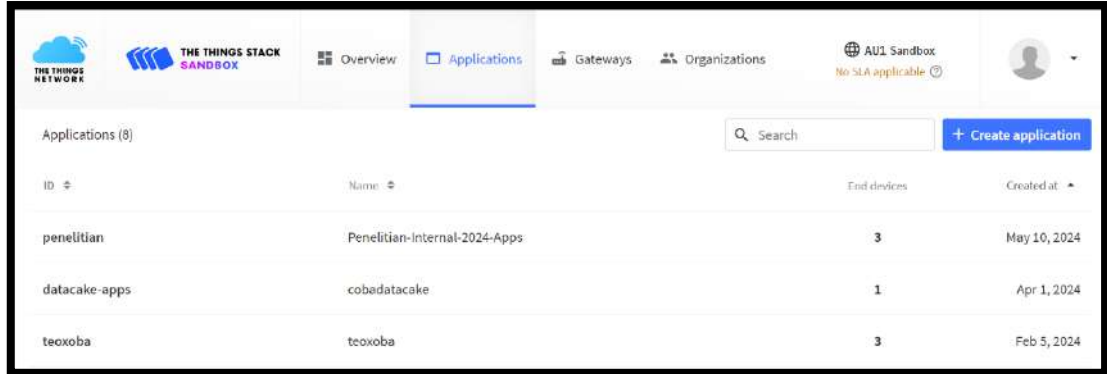


Gambar 3. 5 *Pemilihan cluster TTN*

Pemilihan *cluster* di *The Things Network* (TTN) penting untuk memastikan kinerja dan keandalan jaringan LoRaWAN. Pada Tugas Akhir ini menggunakan *cluster* Australia karena secara geografis, Australia cukup dekat dengan Indonesia, sehingga dapat mengurangi latensi komunikasi dan meningkatkan kecepatan respon jaringan. Australia memiliki regulasi dan standar yang sesuai dengan banyak wilayah di Asia-Pasifik, memastikan kepatuhan terhadap peraturan yang berlaku.

3.1.3.2 Mendaftarkan perangkat *End Node*

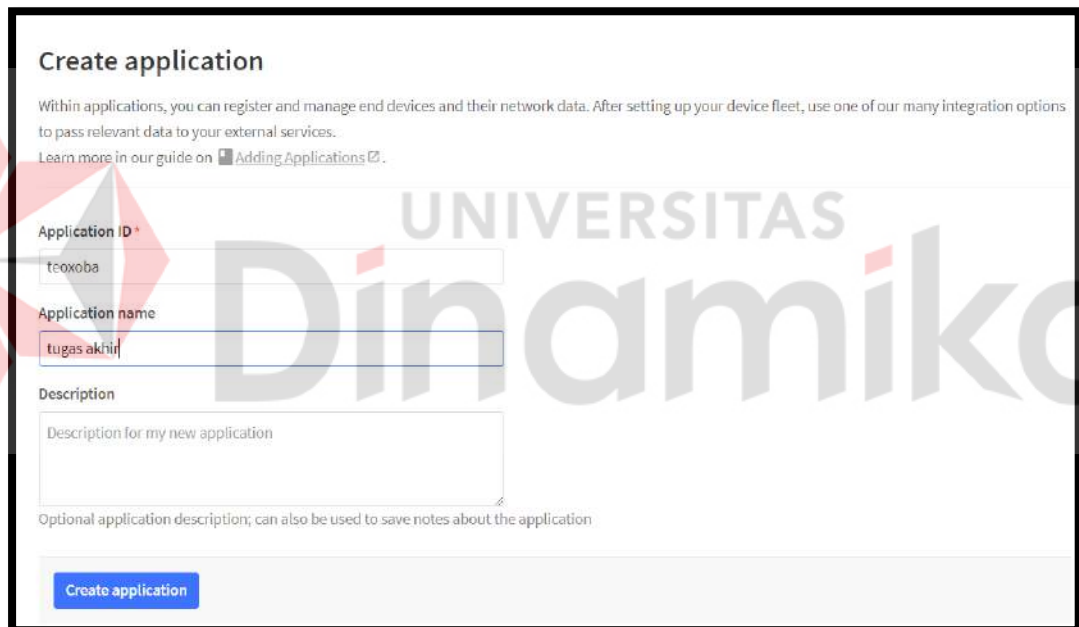
Setelah memilih *cluster* kemudian klik *tab applications* dan klik *create application*.



The screenshot shows the TTN Applications page. At the top, there are navigation tabs: Overview, Applications (selected), Gateways, and Organizations. A search bar and a '+ Create application' button are visible. Below is a table with the following data:

ID	Name	End devices	Created at
penelitian	Penelitian-Internal-2024-Apps	3	May 10, 2024
datacake-apps	cobadatacake	1	Apr 1, 2024
teoxoba	teoxoba	3	Feb 5, 2024

Gambar 3. 6 TTN *application*



The screenshot shows the 'Create application' form. It includes a header, a descriptive paragraph, and several input fields:

- Application ID ***: teoxoba
- Application name**: tugas akhir
- Description**: Description for my new application

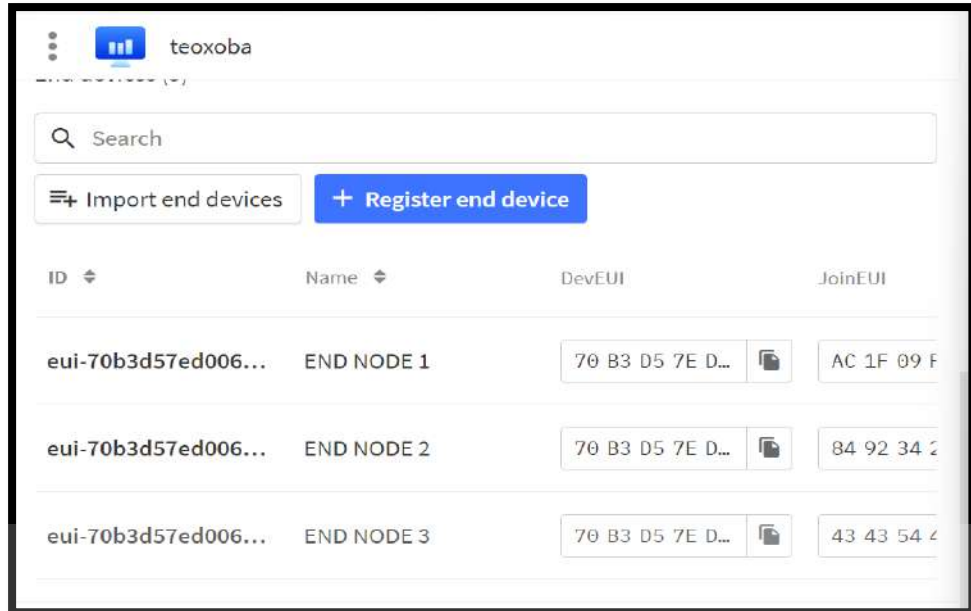
Below the description field, there is a note: 'Optional application description; can also be used to save notes about the application'. At the bottom, there is a 'Create application' button.

Gambar 3. 7 TTN *Create Application*

Pada *tab applications* ini berfungsi untuk mengelola perangkat LoRaWAN, termasuk pendaftaran, pengelompokan, dan manajemen perangkat *end node*. Selain itu juga dapat memfasilitasi pemantauan data dari perangkat *end node*.

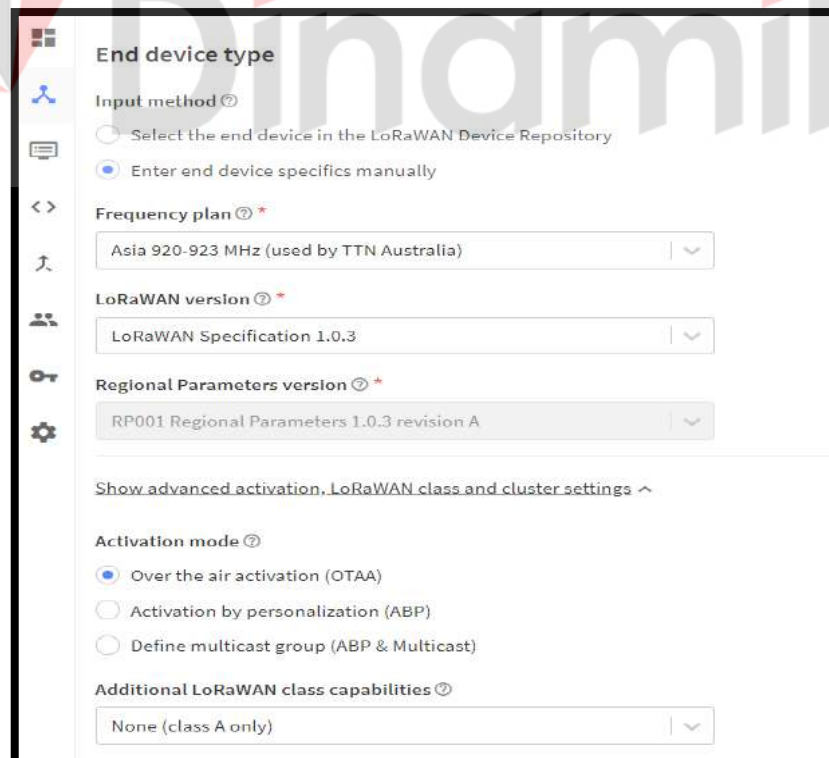
3.1.3.3 Register End Node

Klik pada *applications* yang telah dibuat dan pilih *register end device* untuk mendaftarkan perangkat *end node*.



ID	Name	DevEUI	JoinEUI
eui-70b3d57ed006...	END NODE 1	70 B3 D5 7E D...	AC 1F 09 F...
eui-70b3d57ed006...	END NODE 2	70 B3 D5 7E D...	84 92 34 2...
eui-70b3d57ed006...	END NODE 3	70 B3 D5 7E D...	43 43 54 4...

Gambar 3. 8 TTN *End Decive List*



End device type

Input method

Select the end device in the LoRaWAN Device Repository

Enter end device specifics manually

Frequency plan

Asia 920-923 MHz (used by TTN Australia)

LoRaWAN version

LoRaWAN Specification 1.0.3

Regional Parameters version

RP001 Regional Parameters 1.0.3 revision A

Show advanced activation, LoRaWAN class and cluster settings

Activation mode

Over the air activation (OTAA)

Activation by personalization (ABP)

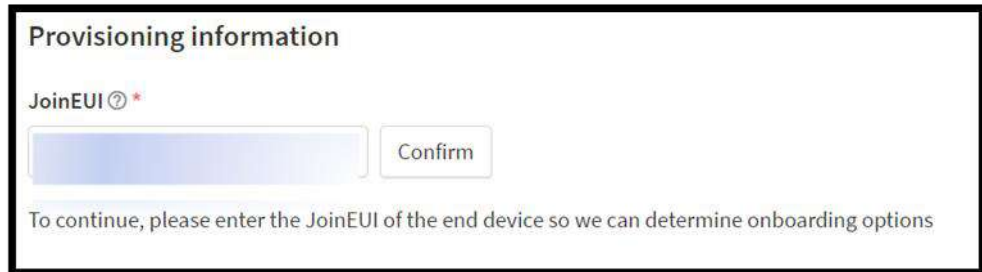
Define multicast group (ABP & Multicast)

Additional LoRaWAN class capabilities

None (class A only)

Gambar 3. 9 TTN *End Device Type*

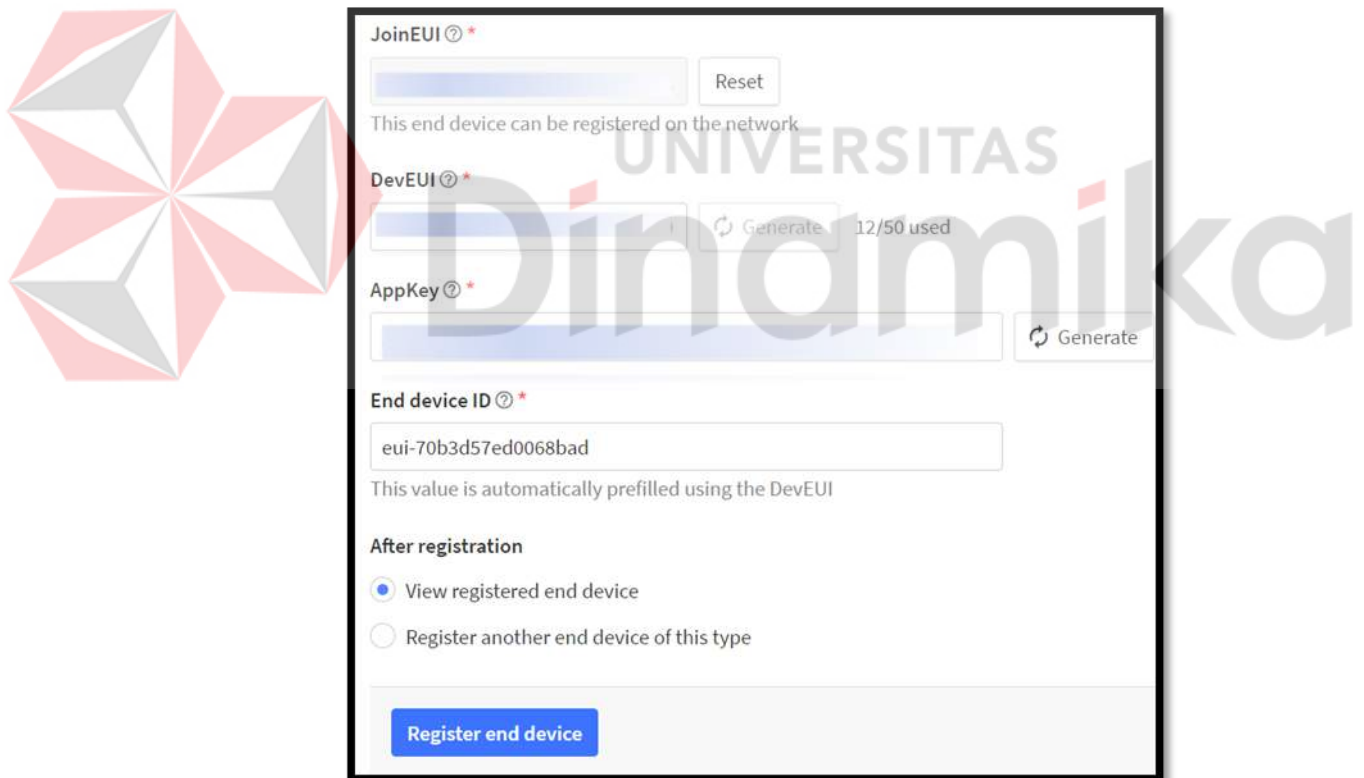
Pengaturan yang digunakan untuk *register* perangkat *end node* adalah manual, jenis frekuensi memakai asia 920-923 MHz karena sesuai dengan aturan ISM antara 923 MHz - 925 MHz. LoRaWAN *version* menggunakan 1.0.3, mode *activation* memakai OTAA, dan jenis class menggunakan *class A*.



The screenshot shows a form titled "Provisioning information". It contains a "JoinEUI" field with a question mark icon and a red asterisk. Below the field is a "Confirm" button. A message below the button reads: "To continue, please enter the JoinEUI of the end device so we can determine onboarding options".

Gambar 3. 10 TTN *Generate Join EUI*

Setelah itu isi Join EUI dengan kode yang unik sebagai pengidentifikasi yang digunakan dalam proses *join* perangkat *end node* ke jaringan LoRaWAN.



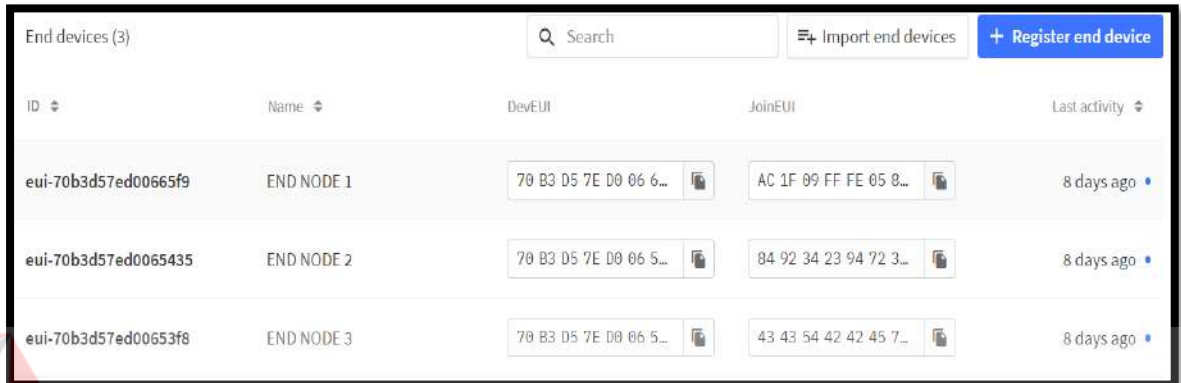
The screenshot shows a form with several fields and buttons. At the top is the "JoinEUI" field with a "Reset" button. Below it is the text "This end device can be registered on the network". The "DevEUI" field has a "Generate" button and shows "12/50 used". The "AppKey" field has a "Generate" button. The "End device ID" field is pre-filled with "eui-70b3d57ed0068bad" and has a note: "This value is automatically prefilled using the DevEUI". Under "After registration", there are two radio buttons: "View registered end device" (selected) and "Register another end device of this type". At the bottom is a blue "Register end device" button.

Gambar 3. 11 TTN *Generate DevEui dan AppKey*

Lalu klik *generate* pada DevEui dan AppKey agar dibuatkan secara otomatis, kemudian klik *register end device*.

3.1.3.4 Payload Formatter

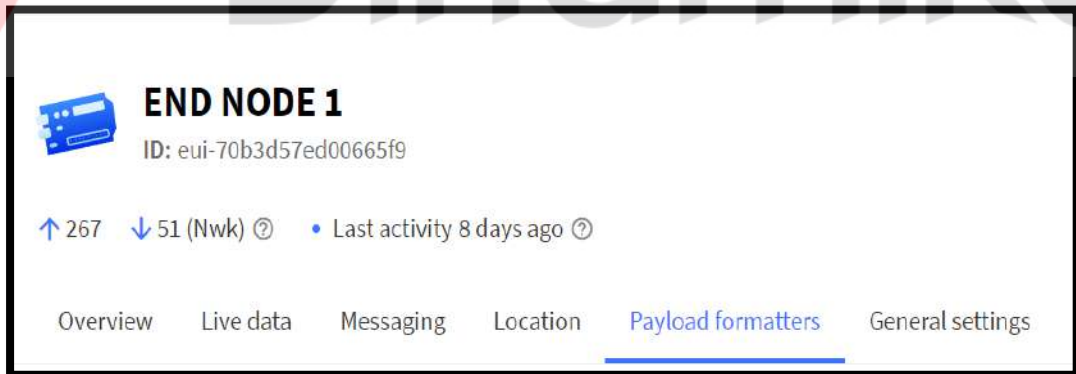
Konfigurasi selanjutnya adalah menguraikan (*decode*) data *payload* yang dikirim oleh perangkat *end node* ke server TTN, karena data yang dikirim oleh perangkat *end node* sering kali dikemas secara terenkripsi untuk efisiensi dan keamanan, sehingga memerlukan proses *decoding* lebih lanjut sebelum data bisa digunakan oleh aplikasi seperti *Datacake*. Pilih *end node* yang telah di *register*.



ID	Name	DevEUI	JoinEUI	Last activity
eui-70b3d57ed00665f9	END NODE 1	70 B3 D5 7E D0 06 6...	AC 1F 09 FF FE 05 8...	8 days ago
eui-70b3d57ed0065435	END NODE 2	70 B3 D5 7E D0 06 5...	84 92 34 23 94 72 3...	8 days ago
eui-70b3d57ed00653f8	END NODE 3	70 B3 D5 7E D0 06 5...	43 43 54 42 42 45 7...	8 days ago

Gambar 3. 12 TTN End Decive List

kemudian pilih tab *payload formatter*, klik tab *uplink* dan *formatter type* menggunakan *javascript formatter*.

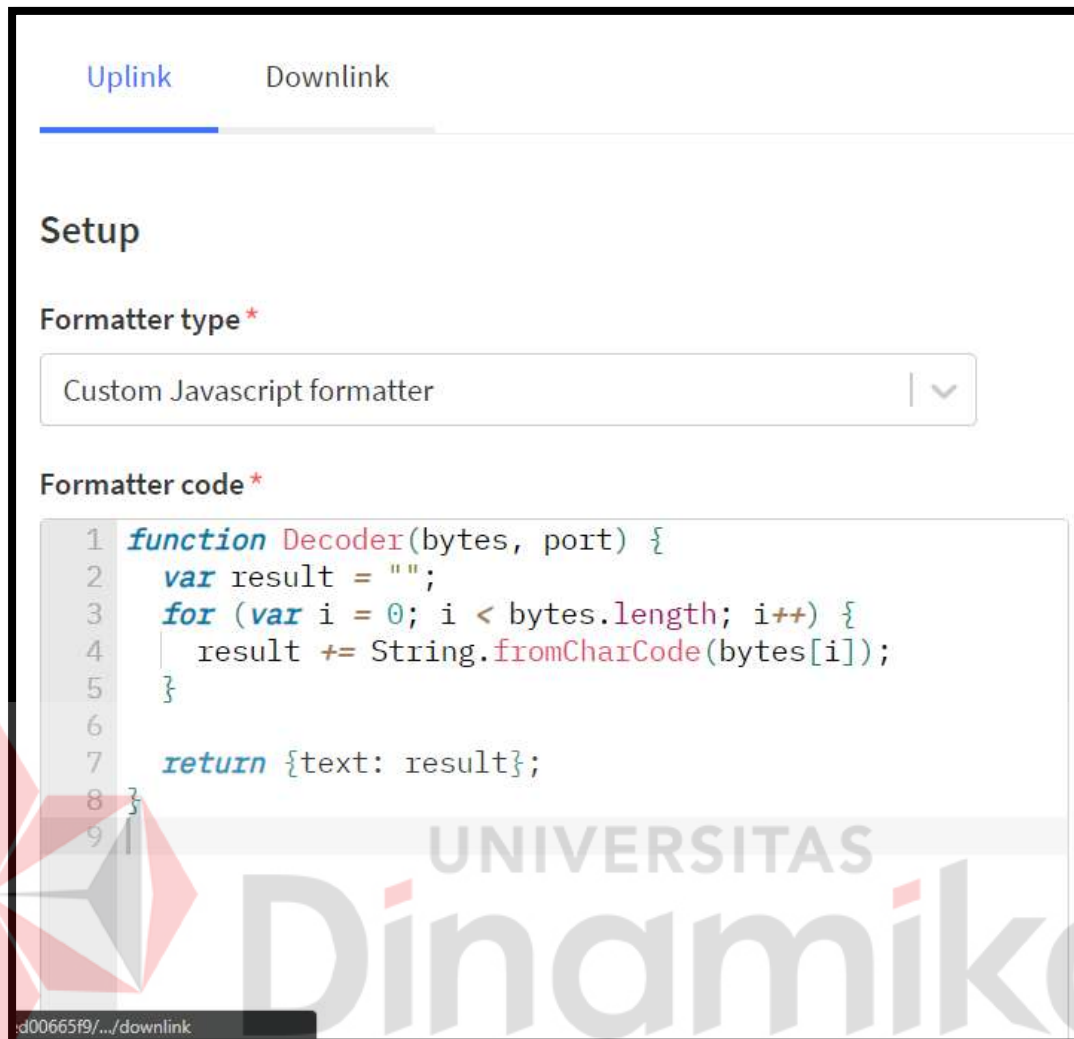


END NODE 1
ID: eui-70b3d57ed00665f9

↑ 267 ↓ 51 (Nwk) Last activity 8 days ago

Overview Live data Messaging Location **Payload formatters** General settings

Gambar 3. 13 TTN End Node Payload Formatter

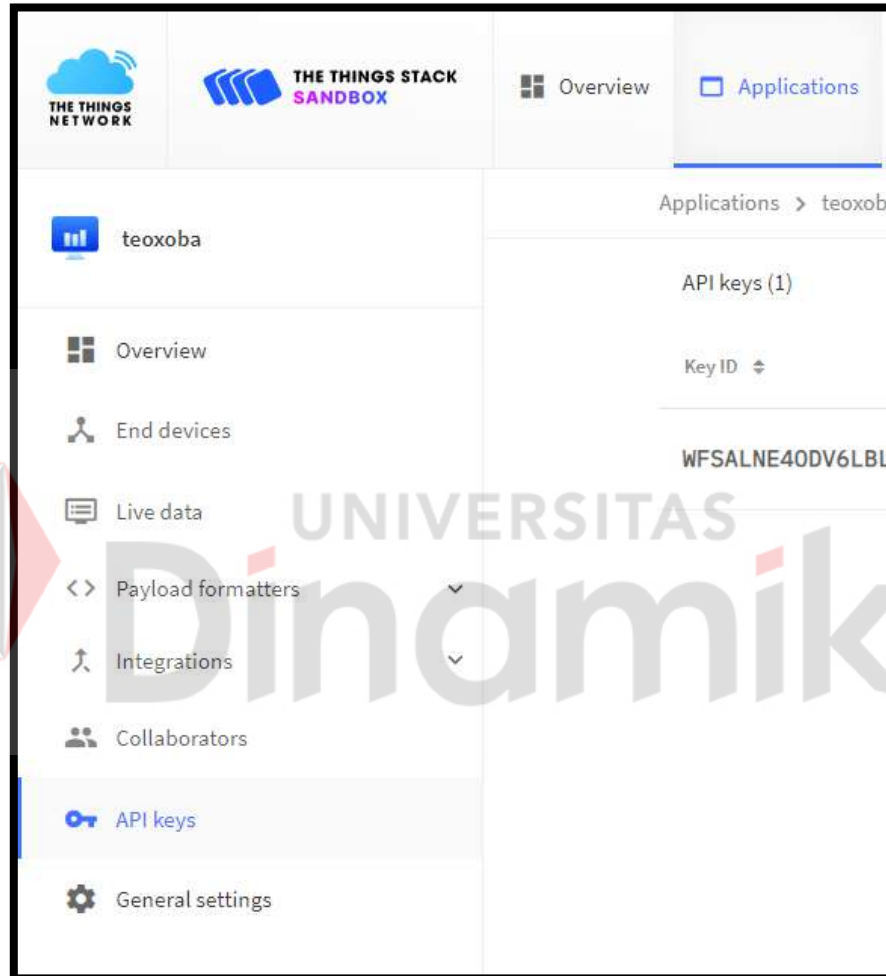


Gambar 3. 14 TTN End Device Formatter Code

Kemudian isi *formatter code* dengan *code decoder* agar data mentah dari format *byte array* menjadi *string*, fungsi ini berguna untuk mengonversi data hasil jarak *haversine* yang masih terenkripsi dalam bentuk *payload* biner menjadi format teks yang lebih mudah dibaca. Lalu klik *save changes*.

3.1.3.5 Intergrations Datacake

Integrasi dengan *Datacake* pada Tugas Akhir ini bertujuan untuk memvisualisasikan dan mengelola data yang telah dikirimkan oleh perangkat *end node* ke server TTN. Untuk mengintegrasikan server TTN dengan *Datacake* masuk ke tab applications yang telah di buat dan klik *API KEYS* Kemudian pilih *add API Key*.



Gambar 3. 15 TTN Application Add API KEYS



Gambar 3. 16 TTN Application API KEYS

Isikan nama dan pilih *grant all current and future rights* dan klik *save changes*. Pembuatan *API Keys* di *tab applications* bertujuan untuk mengamankan dan mengontrol hak akses, memastikan keamanan, dan integritas data.

Applications > teoxoba > API keys > Edit

Edit API key

Key ID *

Name

TEOXOBA-NODE123-API-KEY

Expiry date

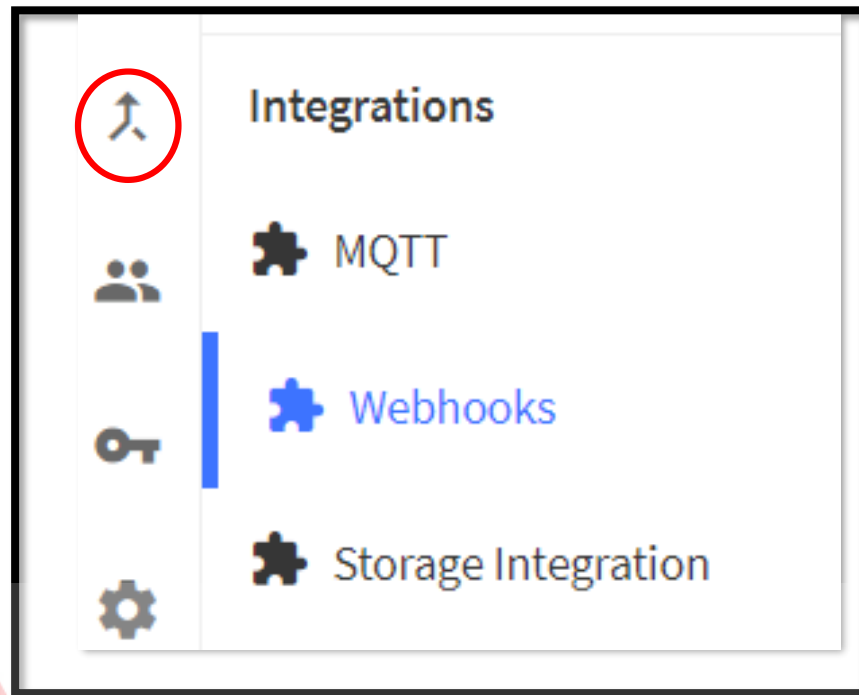
dd/mm/yyyy

Rights *

- Grant all current and future rights
- Grant individual rights
- Select all
 - Delete application
 - View devices in application
 - View device keys in application

Gambar 3. 17 TTN Application Edit API KEY

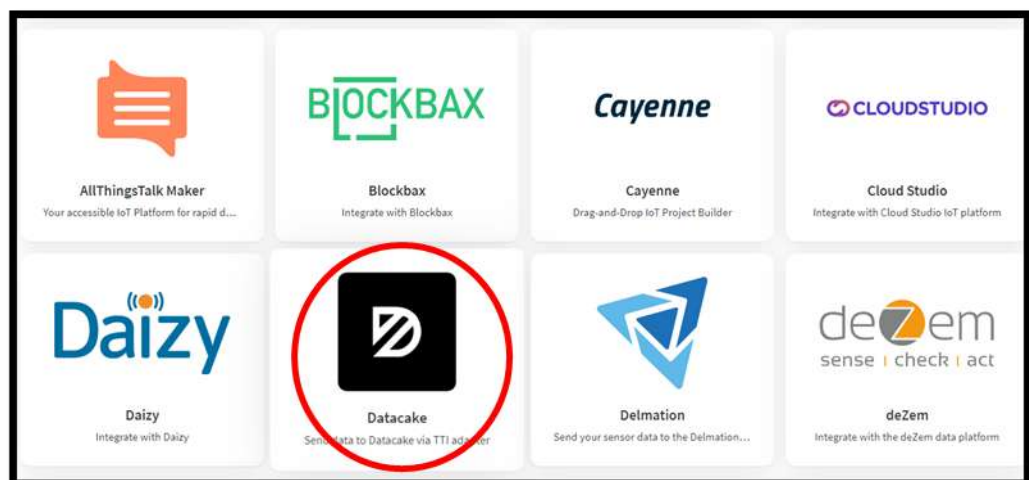
Sesudah membuat *API Key*, pilih tab *applications* klik icon panah segitiga, pilih *webhooks*, klik *add* webhook, dan pilih *Datacake*.



Gambar 3. 18 TTN Application Webhooks

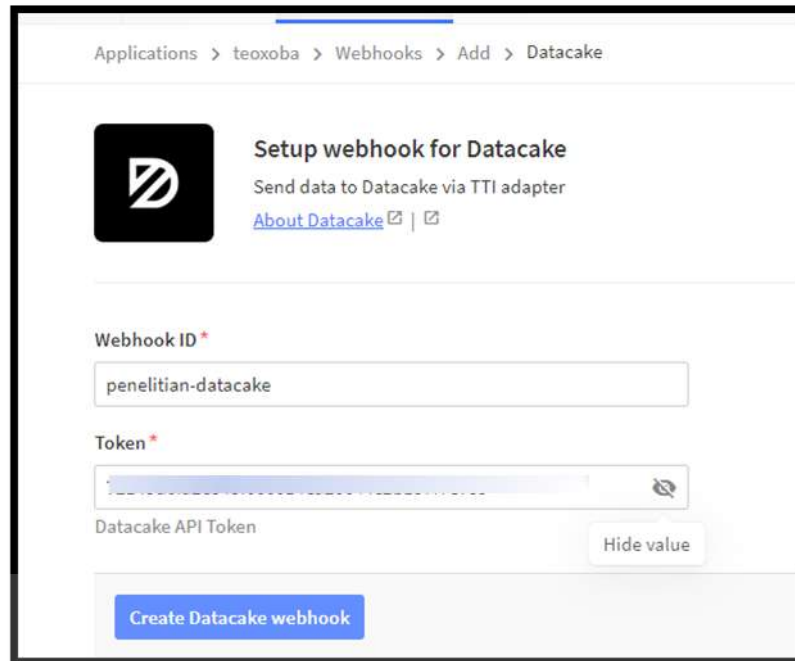


Gambar 3. 19 TTN Application Add Webhooks



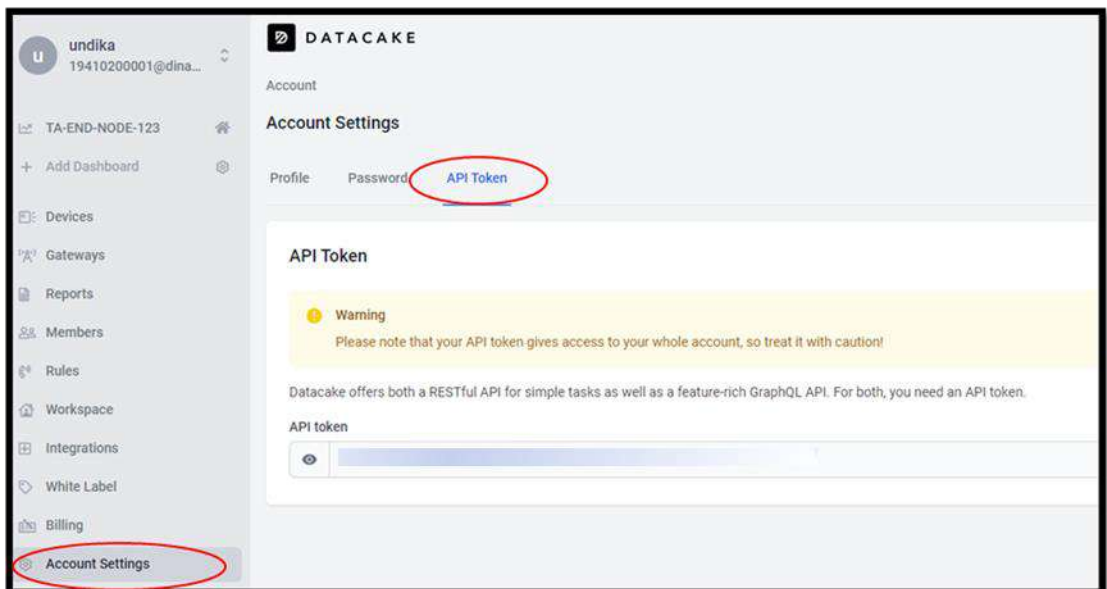
Gambar 3. 20 TTN Add Webhooks Datacake

Kemudian isi *webhook id* dan *API token* dari akun yang telah didaftarkan pada web *Datacake*, klik *create Datacake*.



Gambar 3. 21 TTN *Setup Webhook For Datacake*

Untuk mengetahui *API token Datacake*, terlebih dahulu harus *login* pada akun *Datacake* kemudian masuk ke dashboard utama, pilih *account settings*, tab *Api token* dan *copy*.



Gambar 3. 22 *Datacake API Token*

Edit webhook for Datacake
Send data to Datacake via TTI adapter
[About Datacake](#) | [?](#)

General settings

Webhook ID *

Webhook format *

Base URL *

Downlink API key

The API key will be provided to the endpoint using the "X-Downlink-Apikey" header

Gambar 3. 23 TTN *Edit Webhook For Datacake*

Enabled event types

For each enabled event type an optional path can be defined which will be appended to the base URL

Uplink message /

An uplink message is received by the application

Gambar 3. 24 TTN *Webhook Enabled Event Type*

Kemudian pilih JSON di *webhook format*, dan centang pada *uplink message*, lalu klik *save changes*. Tujuan dari konfigurasi *webhooks* di server TTN, agar *Datacake* dapat menerima data dari perangkat LoRaWAN secara *real-time*.

3.1.4 Datacake

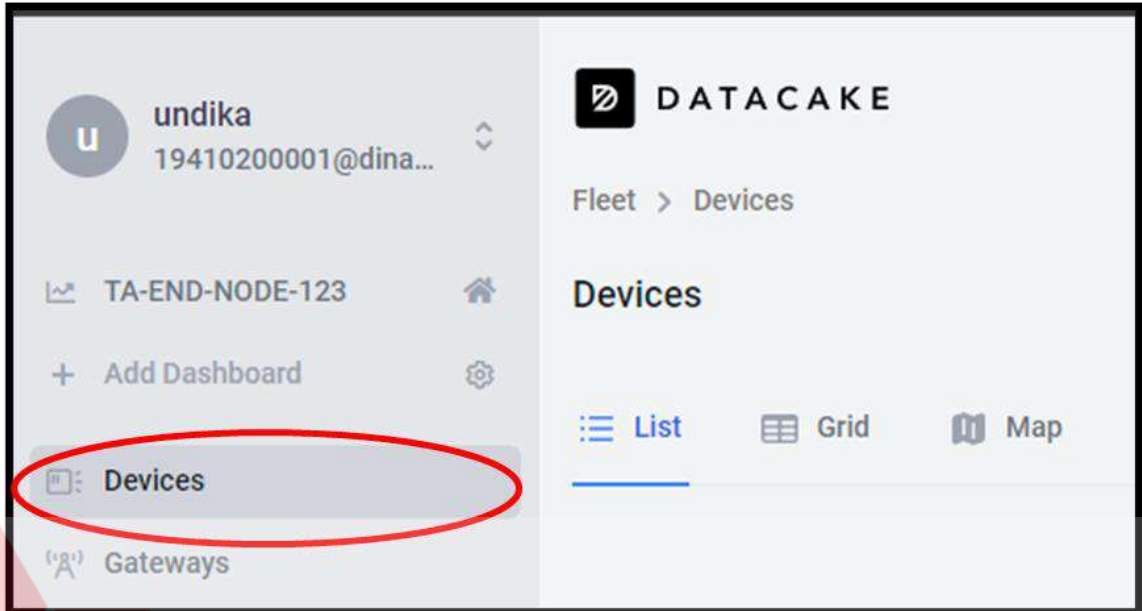
Datacake merupakan bagian akhir yang menampilkan data GPS dan parameter analisis RSSI, SNR, *Spreading Factor*, dan *Datarate*. Oleh karena perlu dilakukan beberapa konfigurasi agar data dari server TTN dapat terhubung ke *Datacake*. Untuk melakukan konfigurasi hal pertama yang perlu dilakukan adalah *Login* ke akun *Datacake*.



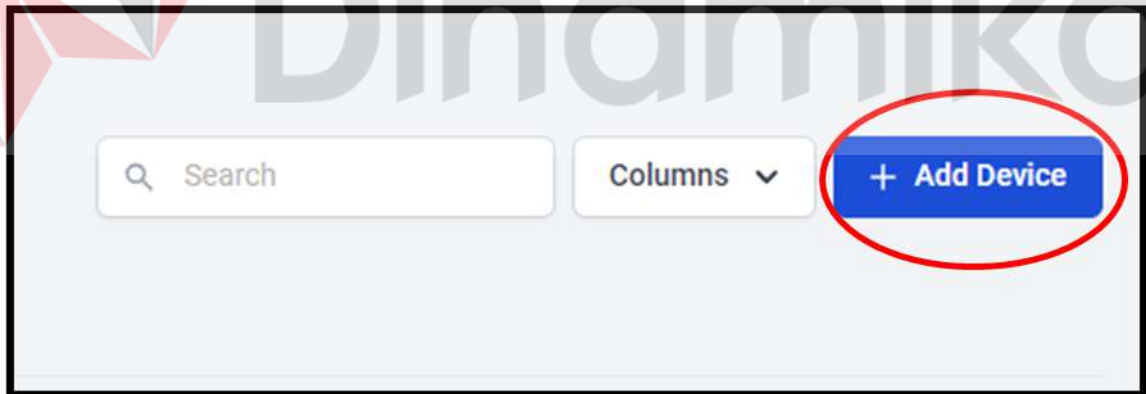
Gambar 3. 25 Login Datacake

3.1.4.1 Add device

Kemudian setelah masuk ke *dashboard* utama, pilih *devices* yang berada di sebelah kiri dashboard utama, dan klik *add device*.

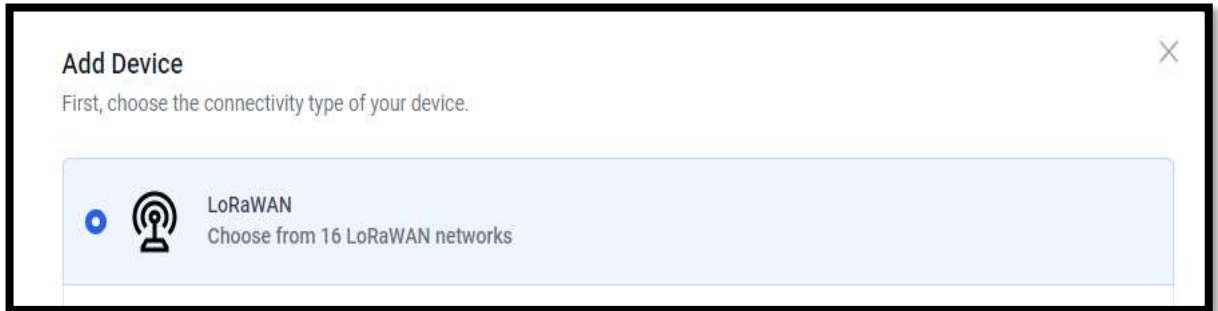


Gambar 3. 26 Datacake Devices



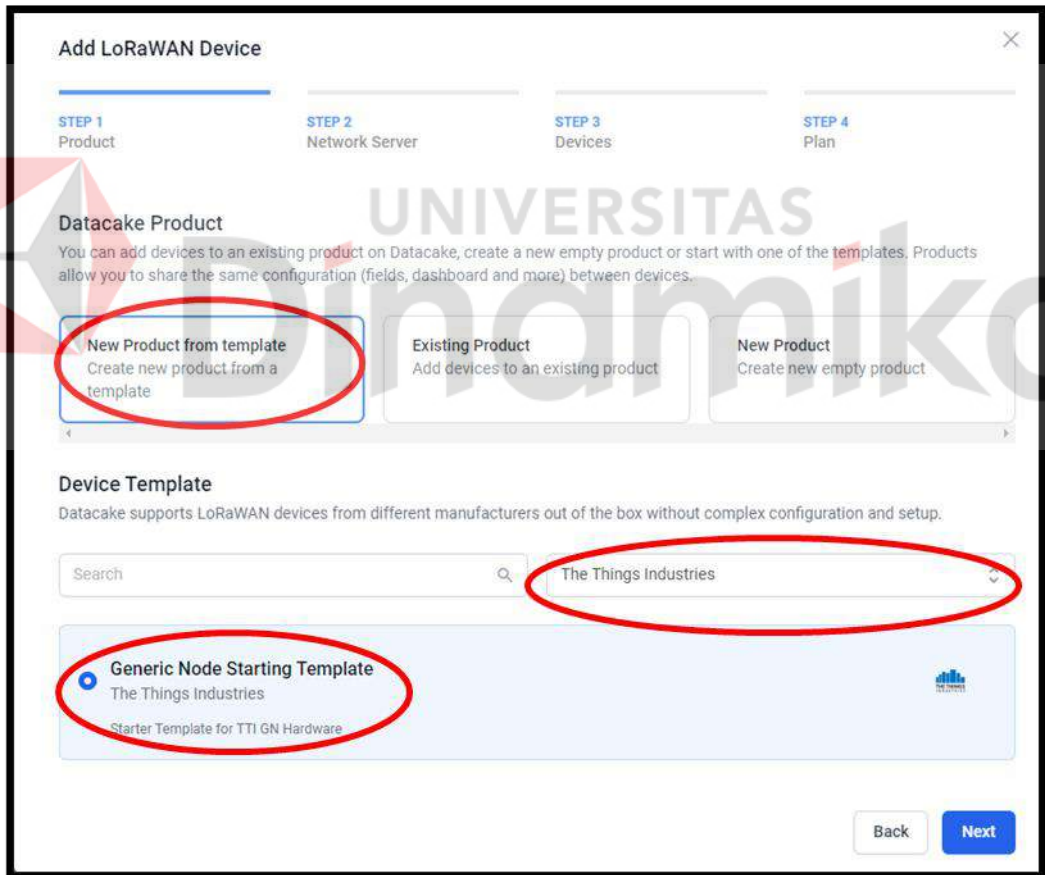
Gambar 3. 27 Datacake Add Device

Setelah itu, pilih tipe LoRaWAN untuk tipe *device* dan klik *next*.



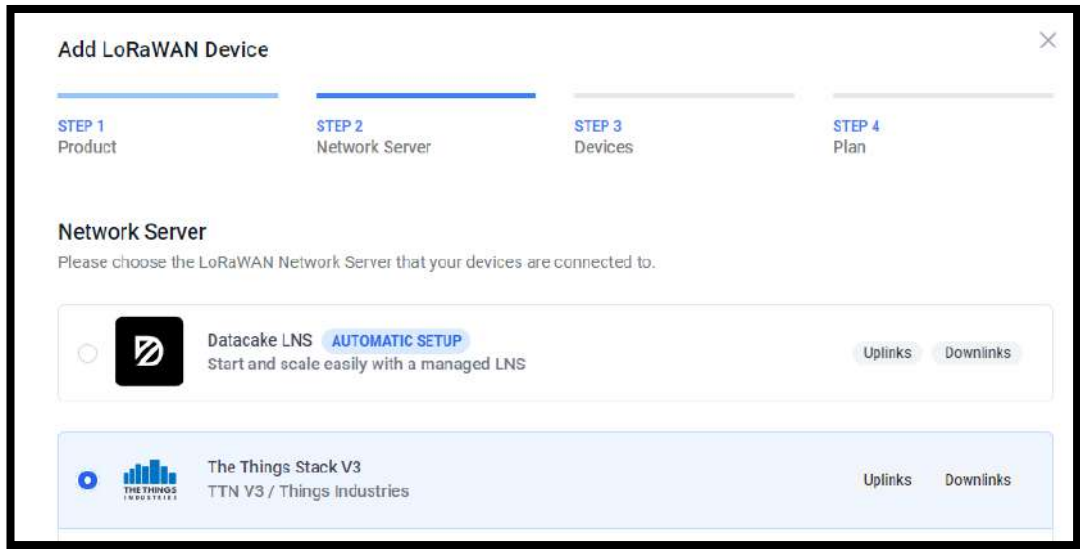
Gambar 3.28 Datacake Add Device Type

Klik *new product from template*, kemudian pilih *the things industries* lalu pilih *generic node starting template*, dan pilih *next*.



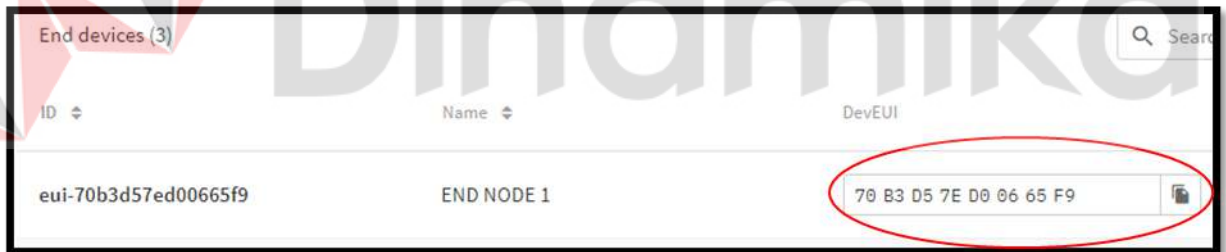
Gambar 3. 29 Datacake Add Device Step 1

Pilih *The Things Stack V3* kemudian klik next.

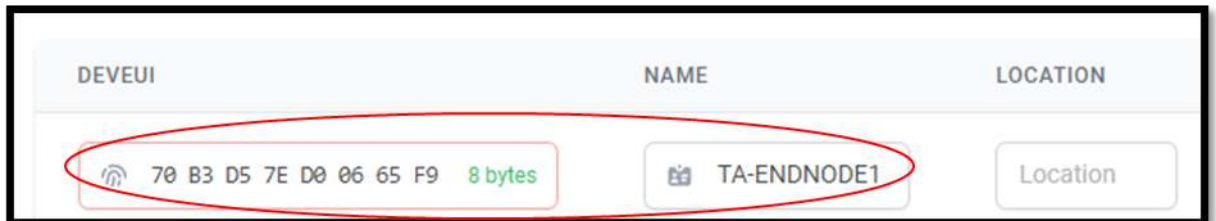


Gambar 3. 30 Datacake Add Device Step 2

Setelah itu, masuk ke Server TTN dan buka *application* yang sudah dibuat sebelumnya, lalu copy DevEUI dari perangkat *end node* dan pastekan pada *DevEUI Datacake*, kemudian isi nama perangkat *end node* dan pilih next.

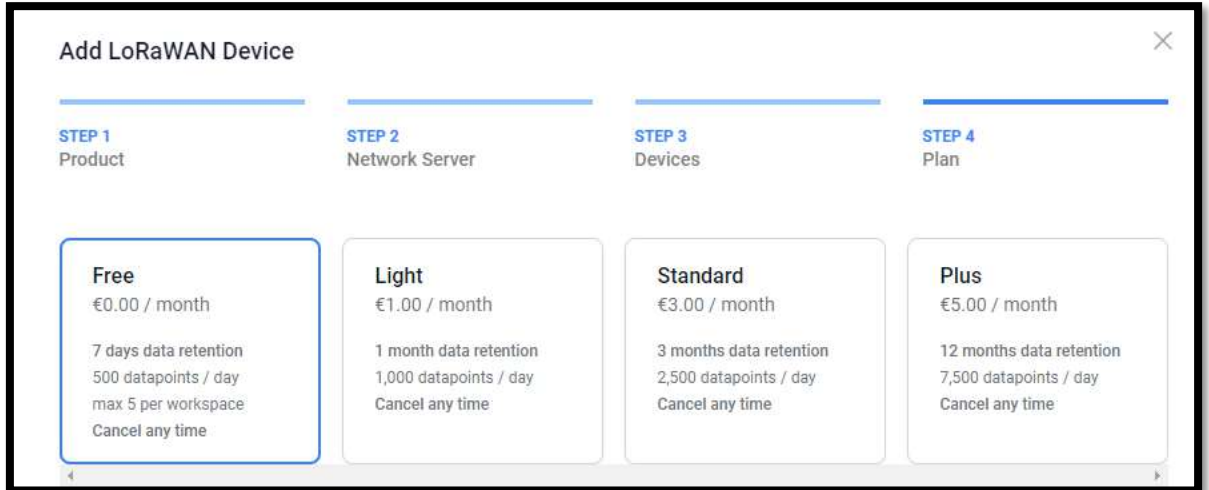


Gambar 3. 31 TTN End Device DevEUI



Gambar 3. 32 Datacake Add Device Step 3

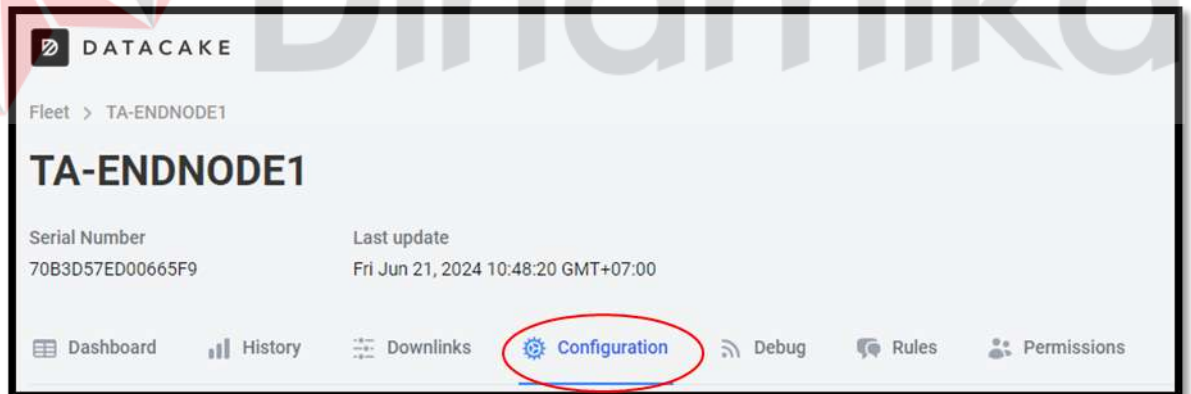
Pilih yang *free* dan klik *add device*, karena akun *Datacake* yang dibuat menggunakan versi gratis. Sehingga maksimal perangkat end node yang bisa di tambahkan hanya sampai 5 perangkat *end node*.



Gambar 3. 33 *Datacake Add Device Step 4*

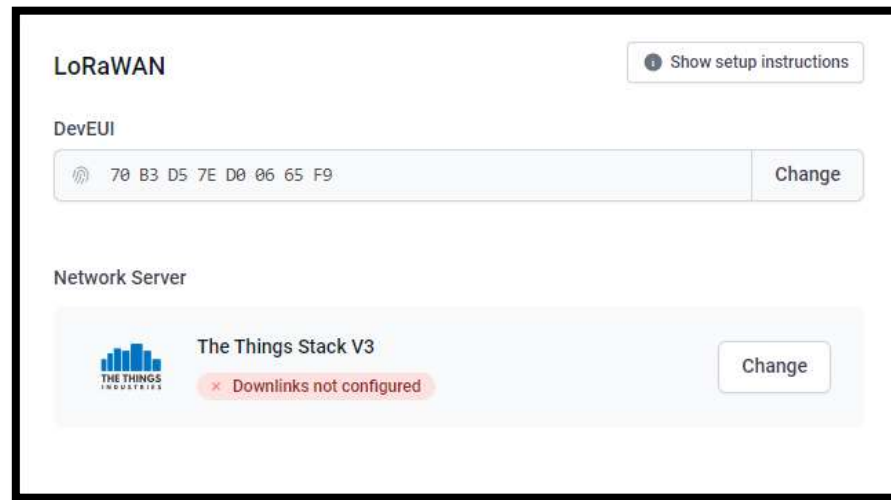
3.1.4.2 Device Configuration

Sesudah melakukan *Add device*, perlu dilakukan *device configuration* dengan cara klik device lalu pilih tab *configuration*.



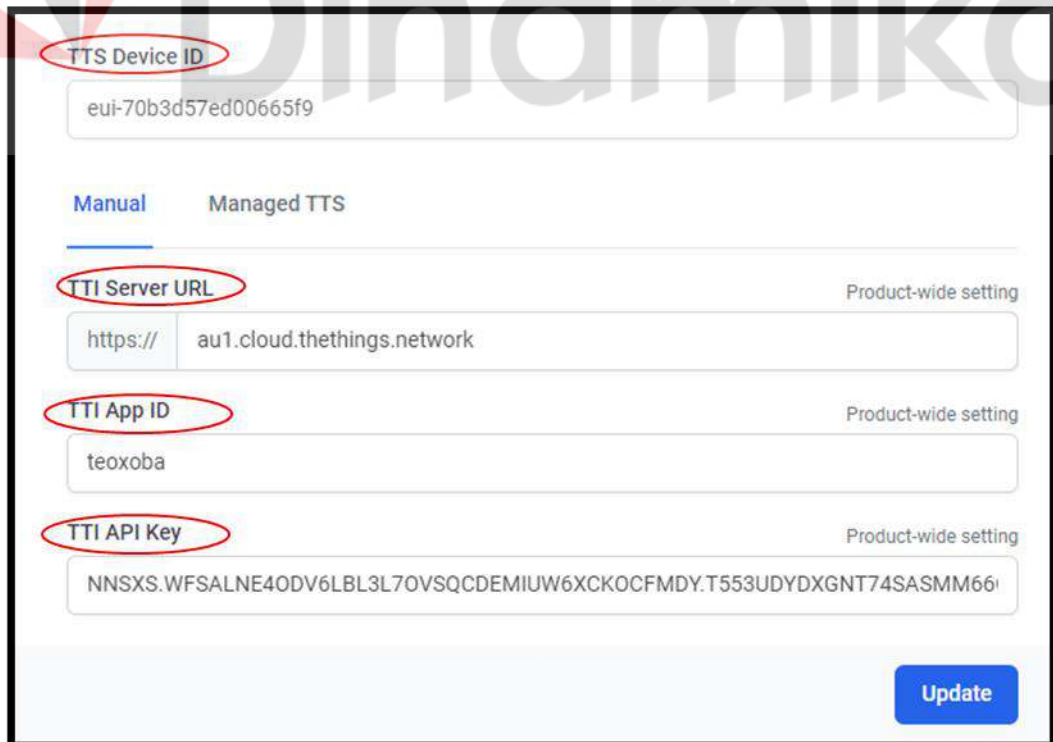
Gambar 3. 34 *Datacake Device Configuration*

Kemudian scroll kebawah hingga kebagian *network server* lalu klik *button change*.



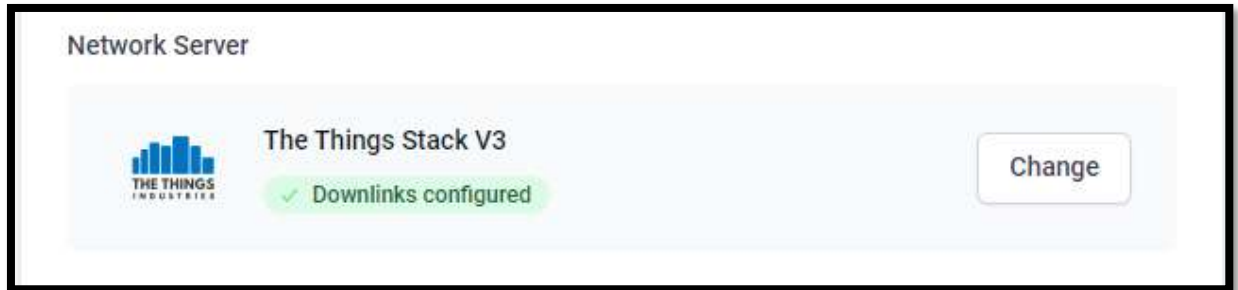
Gambar 3. 35 Datacake Device Downlink Configuration

Pada bagian *TTS Device ID*, *TTI Server URL*, *TTI App ID*, dan *TTI API key* isi sesuai dengan *application* yang telah dibuat pada server TTN, lalu klik *update*.



Gambar 3. 36 Datacake Device Downlink TTI Configuration

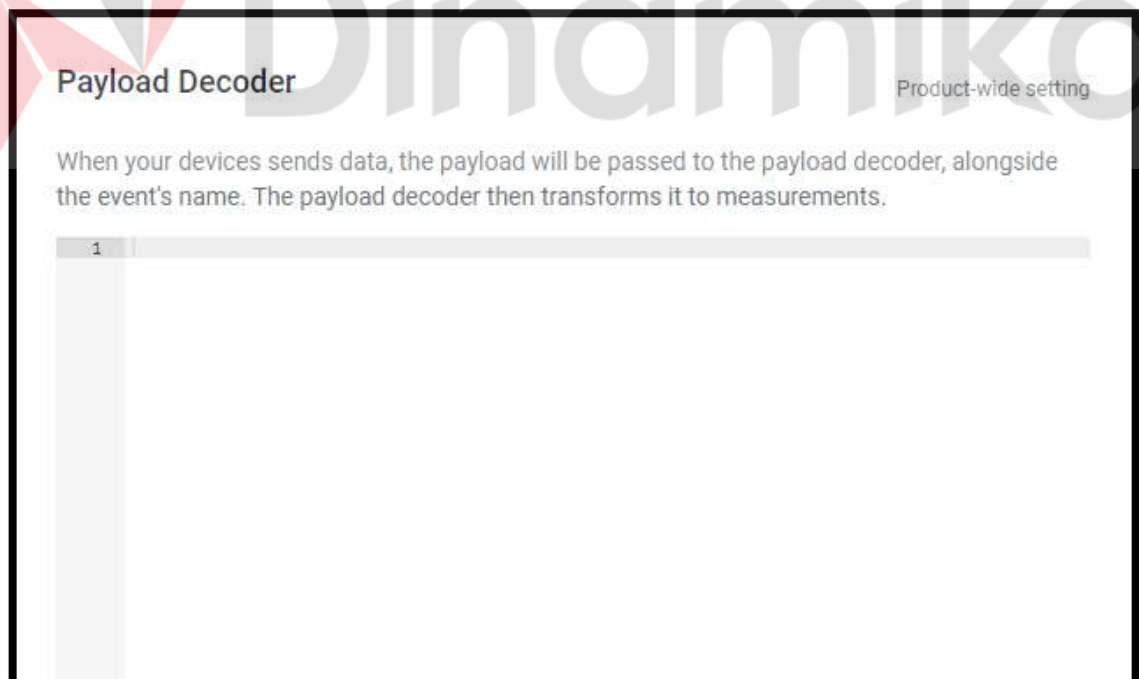
Setelah *update*, maka status *downlink* akan berubah menjadi warna hijau yang berarti *Datacake* berhasil terhubung dengan perangkat *end node* yang sebelumnya di daftarkan pada server TTN.



Gambar 3. 37 *Datacake Device Downlink Configuration*

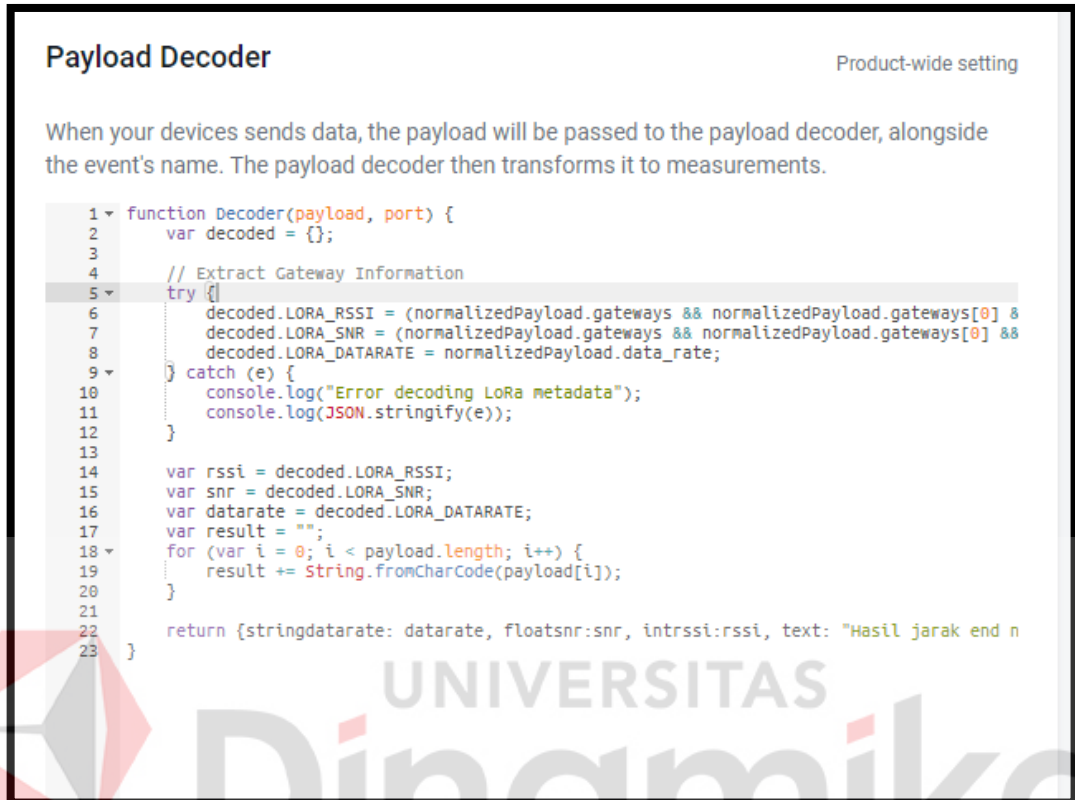
3.1.4.3 *Payload Decoder*

payload decoder pada *Datacake* perlu di lakukan agar untuk mengonversi data mentah ke format yang dapat dibaca dan melakukan *mapping* data agar sesuai dengan *field* yang akan dibuat dengan cara, pilih tab *configuration* dan *scroll* kebawah sampai ke bagian *payload decoder*.



Gambar 3.38 *Datacake Device payload decoder*

Lalu isi *payload decoder* untuk mendekode *payload* dan mengekstrak informasi penting seperti RSSI, SNR, *Spreading Factor*, dan *Data Rate*. Setelah itu klik *save*.



```
1 function Decoder(payload, port) {
2   var decoded = {};
3
4   // Extract Gateway Information
5   try {
6     decoded.LORA_RSSI = (normalizedPayload.gateways && normalizedPayload.gateways[0] &
7     decoded.LORA_SNR = (normalizedPayload.gateways && normalizedPayload.gateways[0] &&
8     decoded.LORA_DATARATE = normalizedPayload.data_rate;
9   } catch (e) {
10    console.log("Error decoding LoRa metadata");
11    console.log(JSON.stringify(e));
12  }
13
14  var rssi = decoded.LORA_RSSI;
15  var snr = decoded.LORA_SNR;
16  var datarate = decoded.LORA_DATARATE;
17  var result = "";
18  for (var i = 0; i < payload.length; i++) {
19    result += String.fromCharCode(payload[i]);
20  }
21
22  return {stringdatarate: datarate, floatsnr:snr, intrssi:rssi, text: "Hasil jarak end n
23 }
```

Gambar 3. 39 Datacake Device payload decoder

Setelah mendekode pada *payload decoder scroll* kebawah hingga sampai bagian *fields* dan klik *add field*, agar data dari perangkat *end node* dari server TTN dapat ditampilkan dengan benar dan dapat dianalisis di *Datacake*.



Gambar 3. 40 Datacake Device Add Field

Isi *type* dan *name* sesuai dengan *payload decocer* yang telah dibuat, kemudian klik *add field*.

```
21  
22 return {stringdatarate: datarate, floatsnr:snr,
```

Gambar 3. 41 *Datacake Device payload decoder*

Add Field ✕

Fields define the schema of the data the device stores.

Type
String

Name
stringdatarate

Identifier
STRINGDATARATE

The field identifier is a unique string that can consist of uppercase letters, numbers and underscores. Once a field has been created, the identifier can not be changed.

Unit Optional
UNIVERSITAS

Role
None

You can define the role of a field, which are unique per product and can be used to add context to global visualisations and reports.

Formula Optional
Formulas can be used to perform calculations on values from other fields. Fields that have a formula can not be written to from a decoder or via the API.

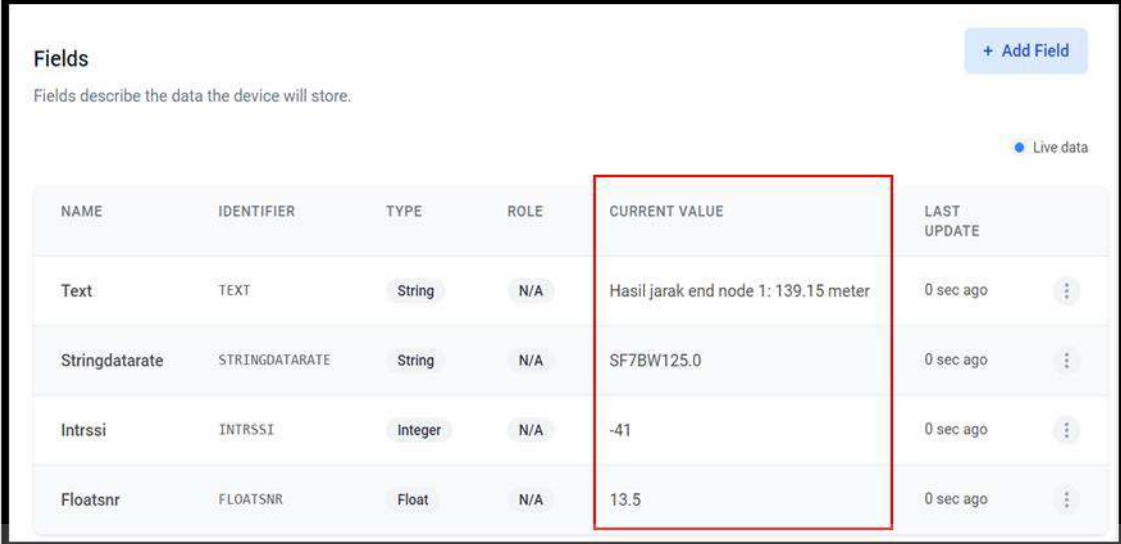
Use Formula

Cancel Add Field

Gambar 3. 42 *Datacake Device Type Field*

Tambahkan field lainnya seperti SNR, RSSI, dan *text*, dengan cara yang sama dan sesuaikan *type* data pada baris *payload decoder* nomor 22.

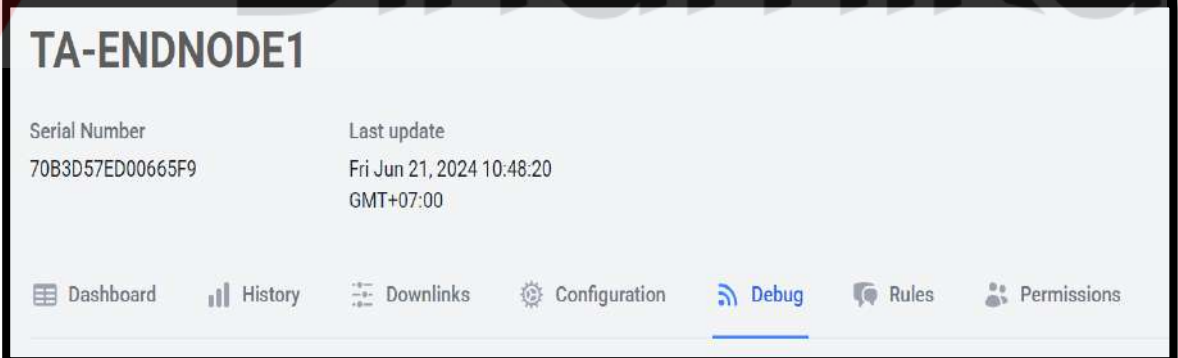
Sesudah semua *field* ditambahkan, maka *current value* akan menampilkan data yang dikirim dari server TTN ke masing-masing *field* yang telah dibuat.



NAME	IDENTIFIER	TYPE	ROLE	CURRENT VALUE	LAST UPDATE
Text	TEXT	String	N/A	Hasil jarak end node 1: 139.15 meter	0 sec ago
Stringdatarate	STRINGDATARATE	String	N/A	SF7BW125.0	0 sec ago
Intrssi	INTRSSI	Integer	N/A	-41	0 sec ago
Floatsnr	FLOATSNR	Float	N/A	13.5	0 sec ago

Gambar 3.43 Datacake Device List Field

Untuk memastikan apakah data dari server TTN berhasil masuk, pilih *device*, lalu klik tab *debug*.



TA-ENDNODE1

Serial Number: 70B3D57ED00665F9
Last update: Fri Jun 21, 2024 10:48:20 GMT+07:00

Dashboard | History | Downlinks | Configuration | **Debug** | Rules | Permissions

Gambar 3.44 Datacake Device Debug

klik *received webhook data from tti*, jika menampilkan data JSON maka server TTN berhasil diterima oleh *Datacake*, dan untuk memastikan apakah masing – masing fields berhasil mengekstrak data dari RSSI, SNR, *Data rate*, dan *text*, klik *recorded measurements from payload decoder*.

Debug Log		
Time	Title	Details
6/21/2024, 10:48:20 AM	Recorded measurements from payload decoder	Decoder returned: {"floatsnr": 13.5, "intrssi": -41, "s
6/21/2024, 10:48:20 AM	Received webhook data from tti	{"end_device_ids":{"device_id":"eui-70b3d57ed00665f9

Gambar 3. 45 *Datacake Device Debug Log*



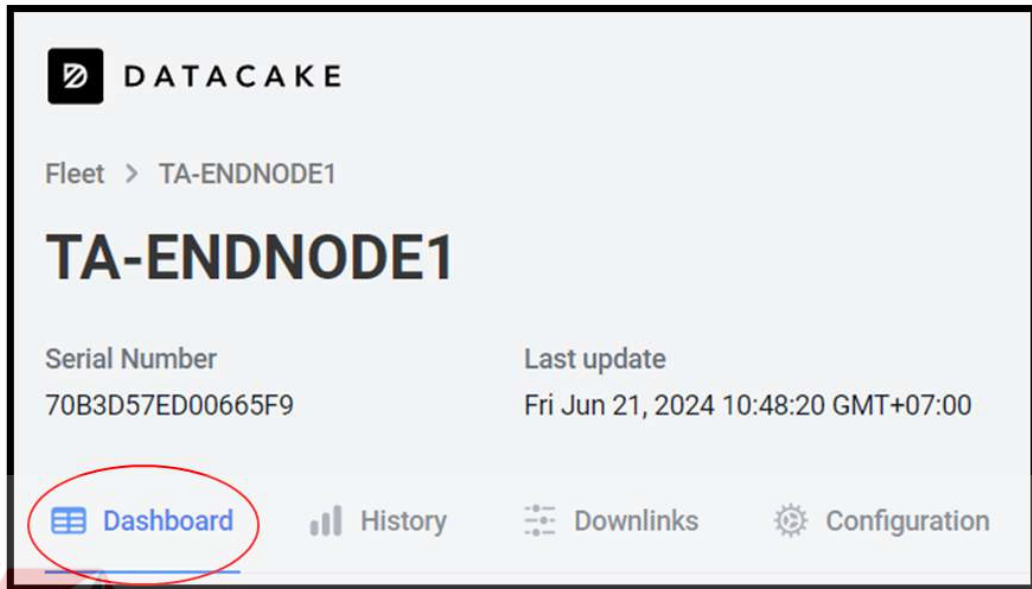
Gambar 3. 46 *Datacake Device Debug Payload Decoder*



Gambar 3. 47 *Datacake Device Debug JSON TTN*

3.1.4.4 Dashboard Device

Konfigurasi terahir yang perlu dilakukan adalah mengatur dashboard visualisasi agar dapat menampilkan data dengan baik dan menarik. Pilih *device* dan klik tab dashboard.



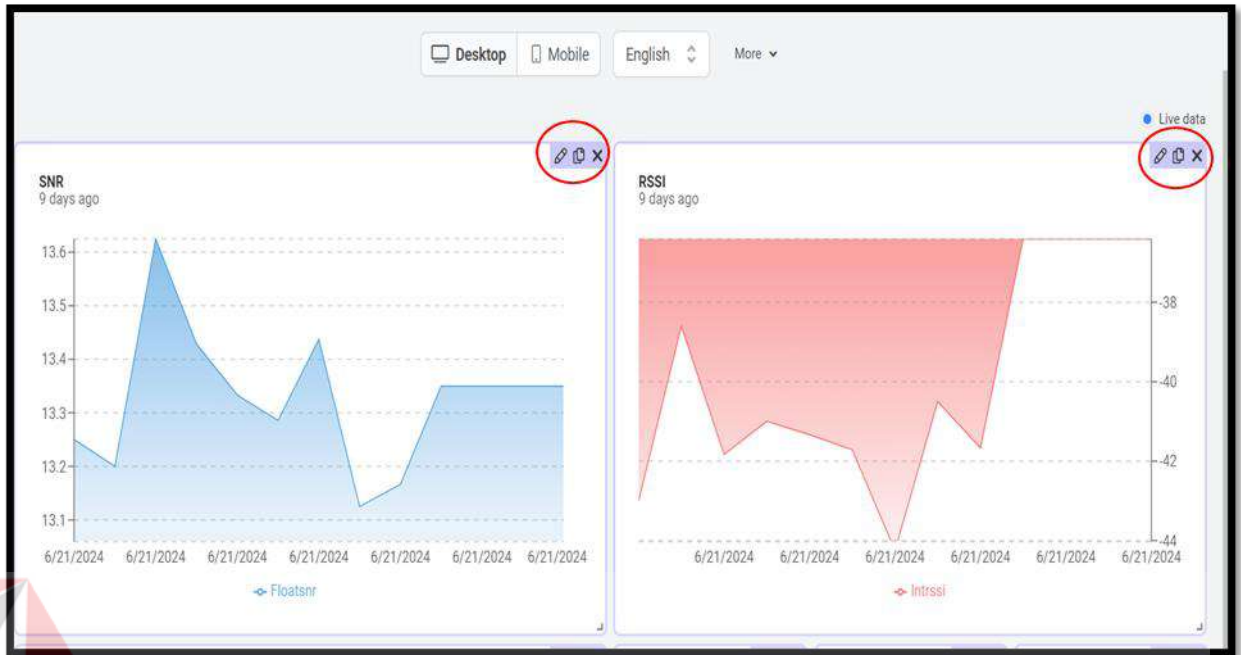
Gambar 3. 48 Datacake Device Dashboard

Kemudian aktifkan button yang berada di pojok kanan.



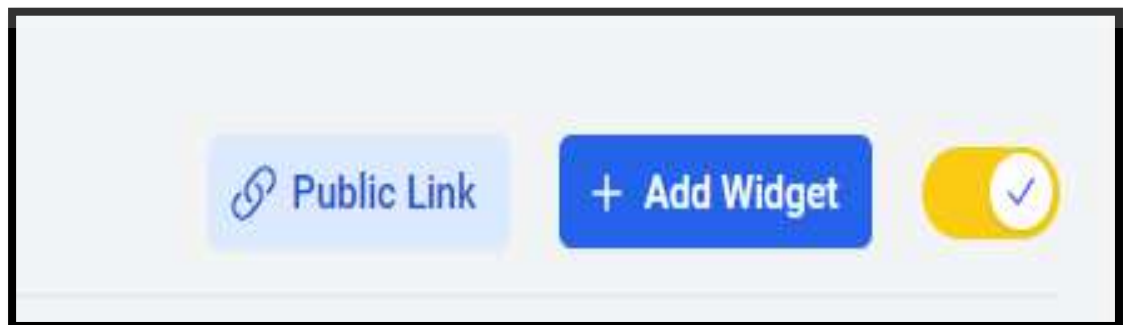
Gambar 3. 49 Datacake Device Enable Edit

Hapus *widget default* dengan cara klik tanda silang pada pojok kanan atas.



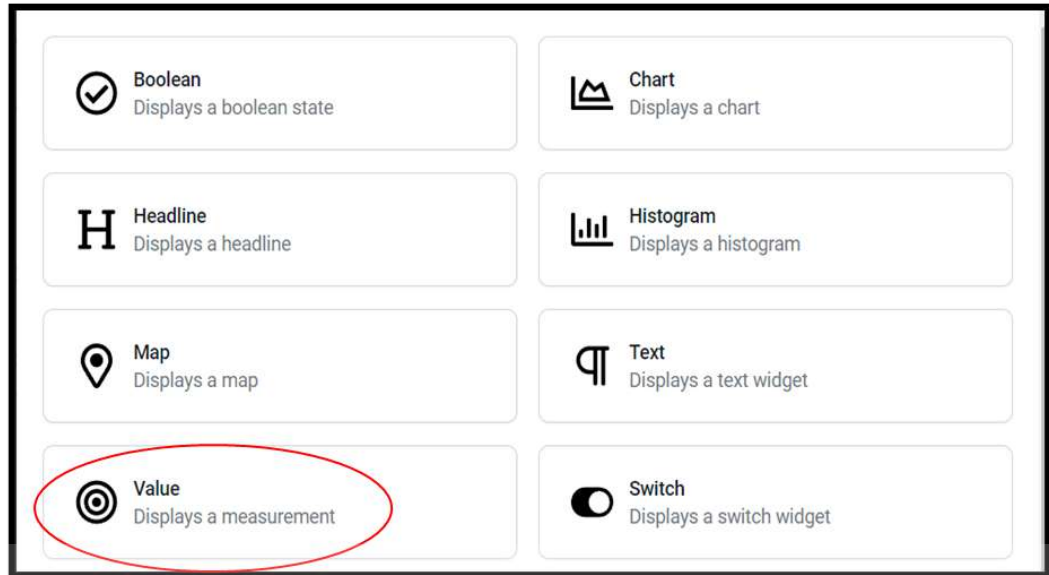
Gambar 3. 50 Datacake Device Widget

Setelah menghapus *widget default*, klik *add widget* untuk membuat *widget* baru.



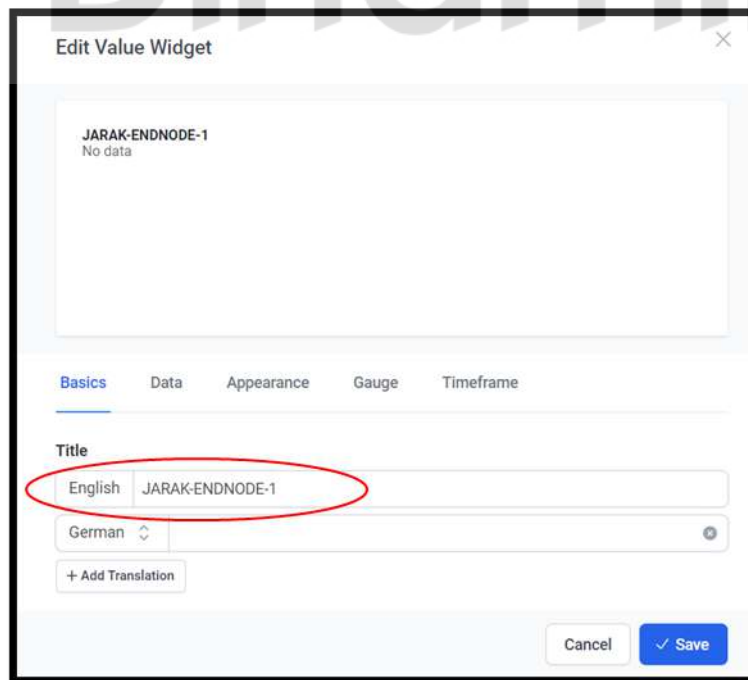
Gambar 3. 51 Datacake Device Add Widget

Pilih tipe *widget* yang ingin ditambahkan, seperti *value* untuk menampilkan data hasil jarak *haversine* dari perangkat *end node*.



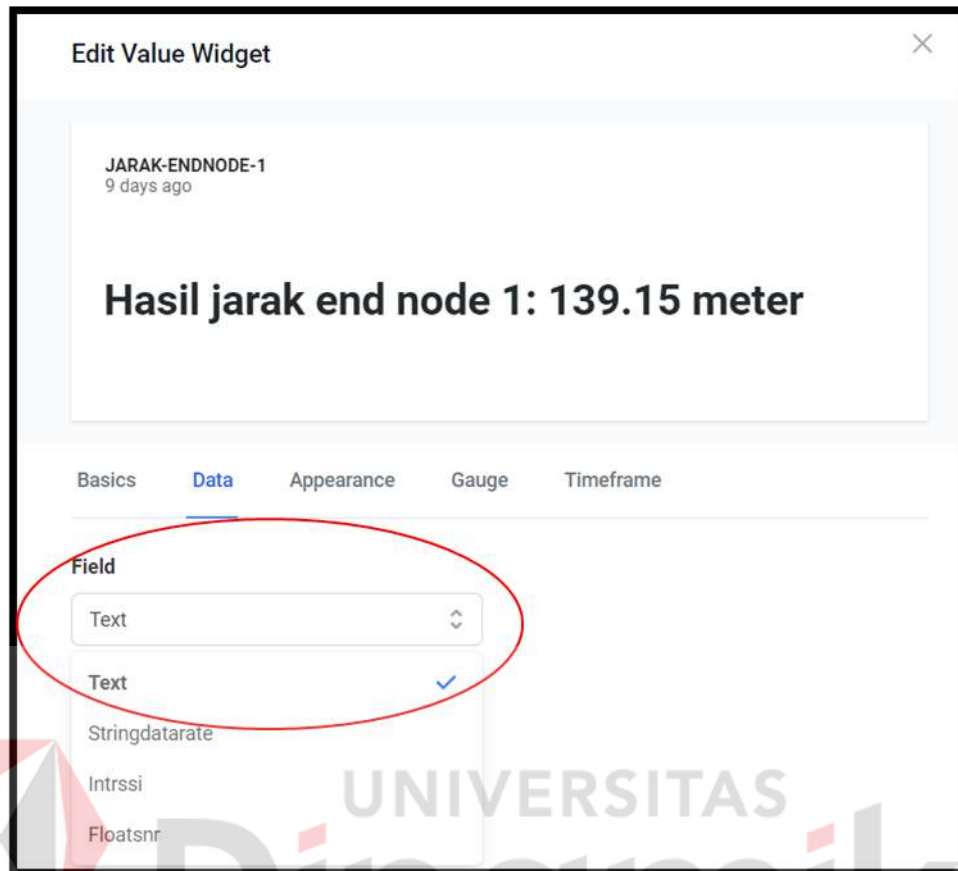
Gambar 3. 52 Datacake Device Widget Type

Isi *title* dengan nama jarak-*endnode-1*, kemudian klik tab *data* untuk menambahkan *field*.



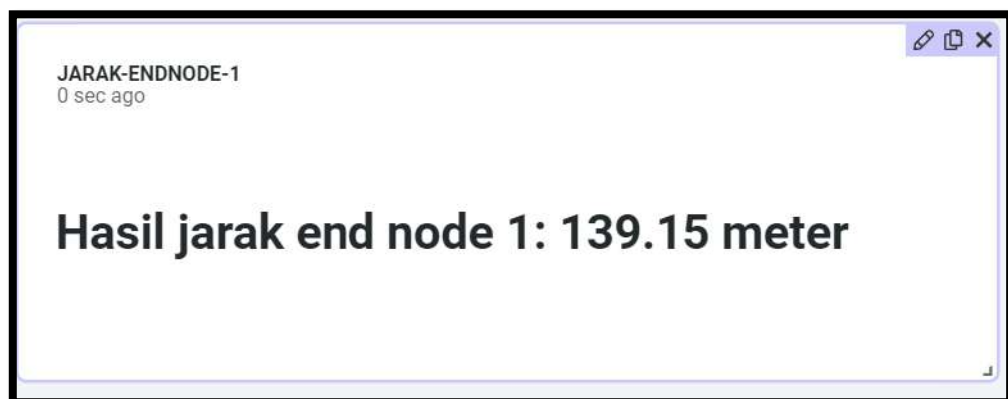
Gambar 3. 53 Datacake Device Widget Title

Pada tab data klik bagian *field*, dan pilih text lalu klik *save*.



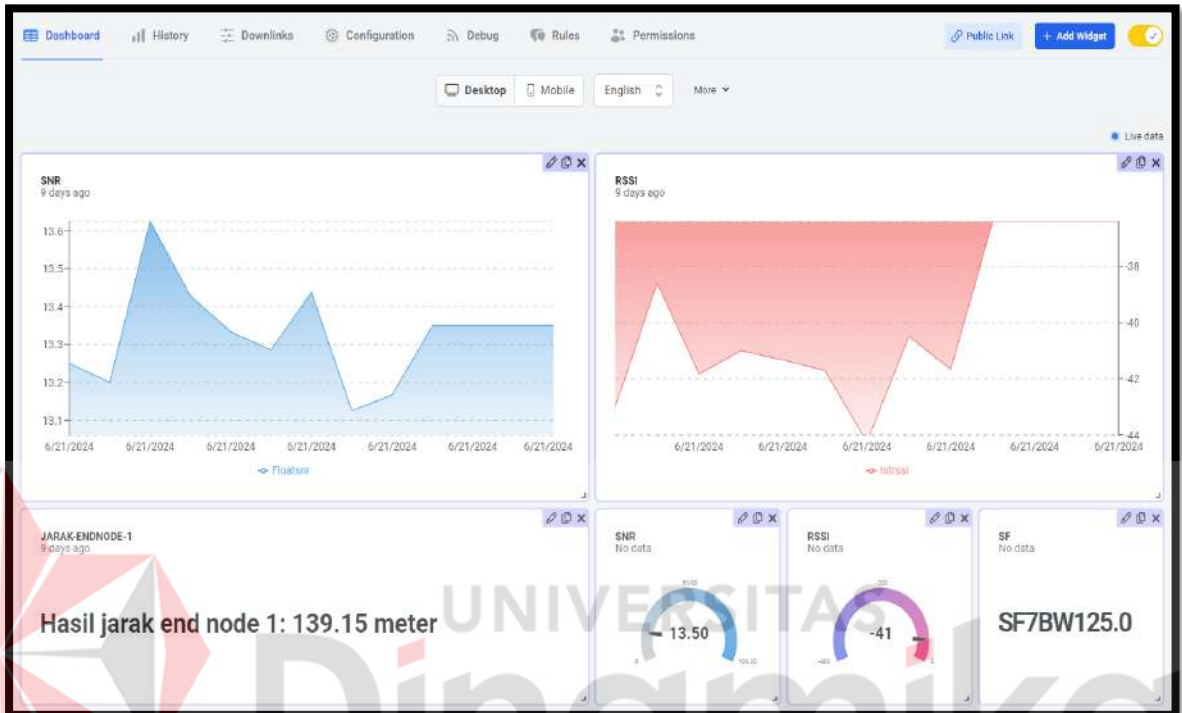
Gambar 3. 54 *Datacake Device Widget Field*

Susudah save maka widget akan tampil pada dashboard, tambahkan juga widget untuk RSSI, SNR, dan *data rate*. Untuk tipe *widget* dapat dipilih jenisnya sesuai dengan kebutuhan.

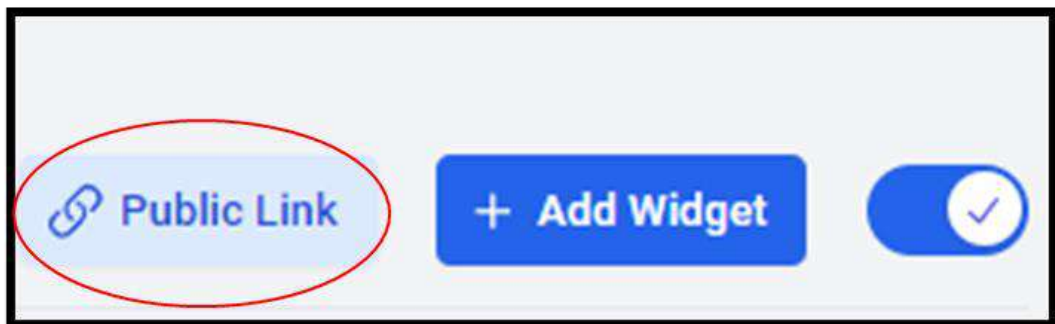


Gambar 3. 55 *Datacake Device Widget*

Gambar 3.56 menampilkan hasil *widget* yang sudah dibuat, Langkah selanjutnya perlu mengaktifkan *public link* agar dapat membagi *dashboard visual* ke perangkat *mobile* dan *desktop* untuk memonitoring data secara *real-time*. Klik *public link* seperti contoh gambar 3.57.

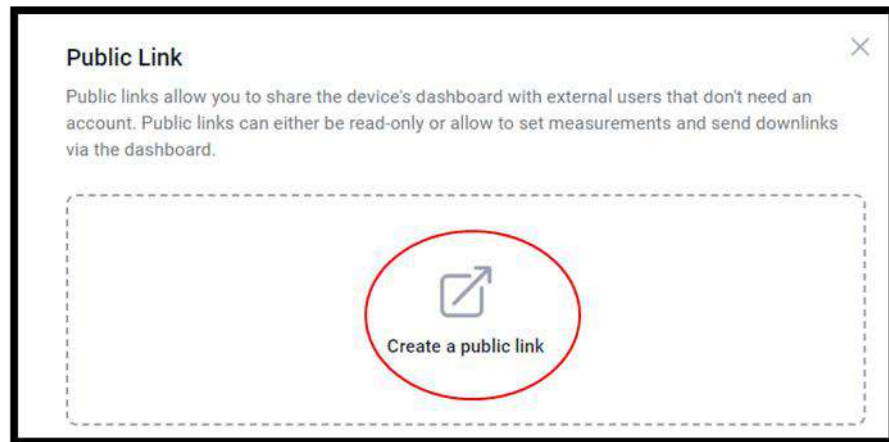


Gambar 3. 56 Datacake Device Widget



Gambar 3. 57 Datacake Device Public Link

klik create a *public link*, pilih mode *read only* pada gambar 3. Dan klik *create public link*.

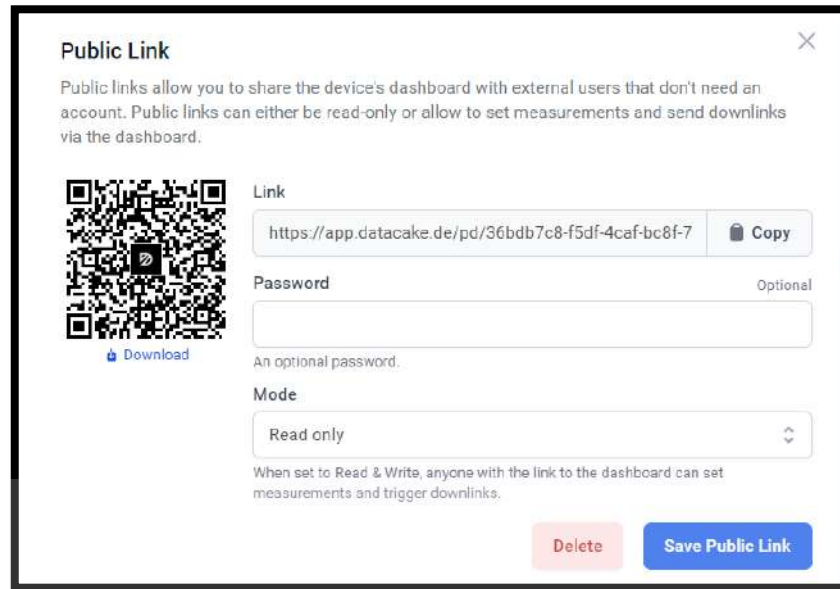


Gambar 3. 58 Datacake Device Create Public Link



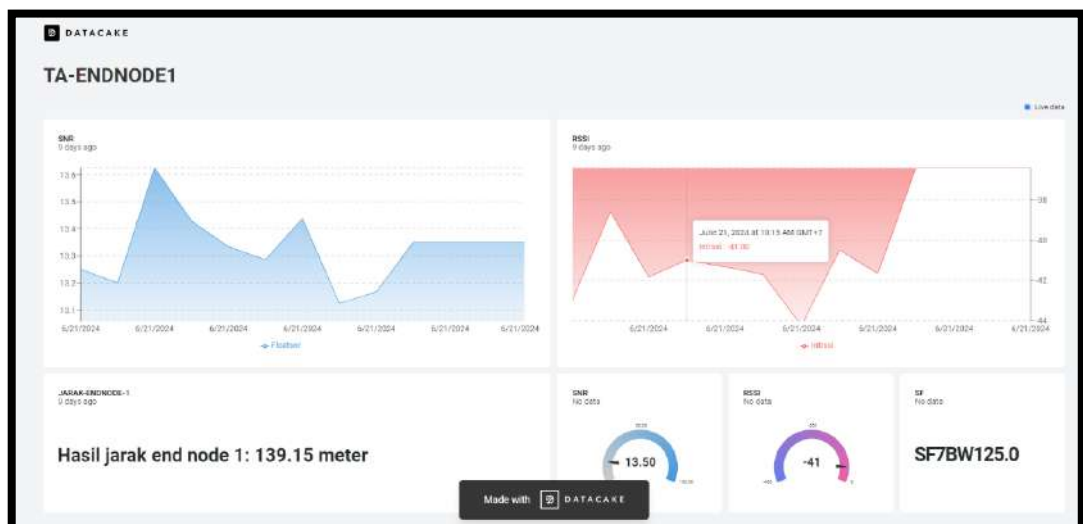
Gambar 3. 59 Datacake Device Public Link Mode

Sesudah *create public link*, akan muncul *barcode* dan *link* yang bisa dibagikan kepada perangkat seperti *mobile* dan *desktop*. Copy *link* lalu klik *save public link*.



Gambar 3. 60 Datacake Device Save Public Link

Paste *link* pada *web browser*, maka akan menampilkan dashboard visualisasi yang telah dibuat.



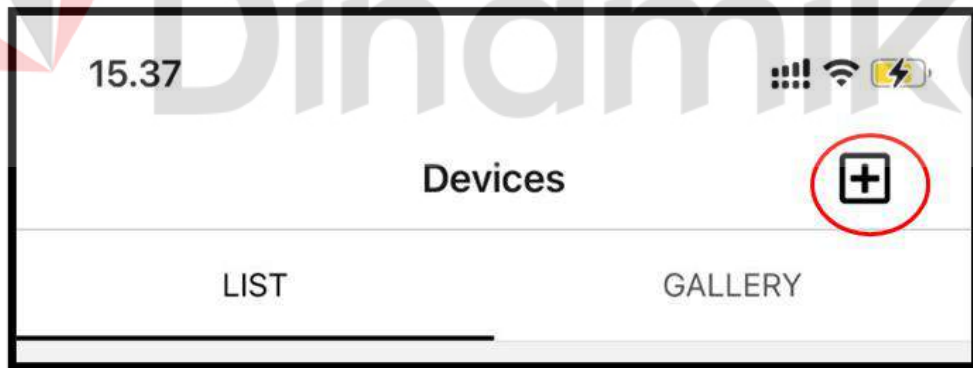
Gambar 3. 61 Datacake Device Dashboard visual

Selain itu cara untuk menambahkan dashboard visualisasi pada perangkat *mobile* bisa menggunakan aplikasi *Datacake* yang tersedia di *appstore* dan *playstore*. Buka aplikasi *Datacake* pada perangkat *mobile*.



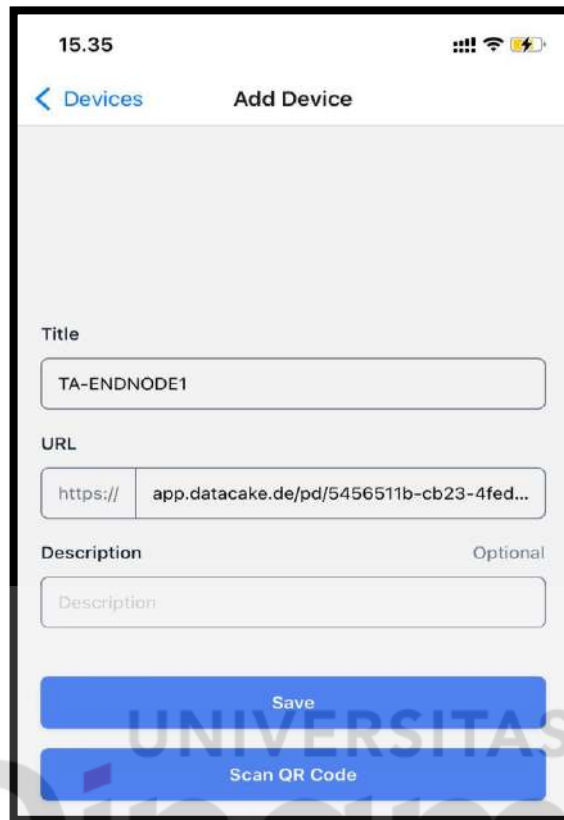
Gambar 3. 62 *Datacake* Aplikasi Handphone

Setelah itu klik icon + pada pojok kanan atas untuk menambahkan perangkat *end node*.



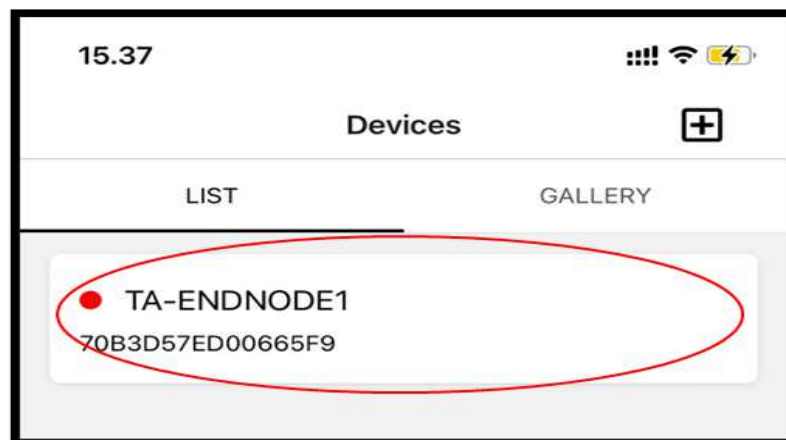
Gambar 3. 63 *Add Device Datacake* dari Aplikasi Handphone

Isi title dan pastekan *public link* yang dibuat sebelumnya atau menggunakan *scan barcode*, lalu pilih *save*.



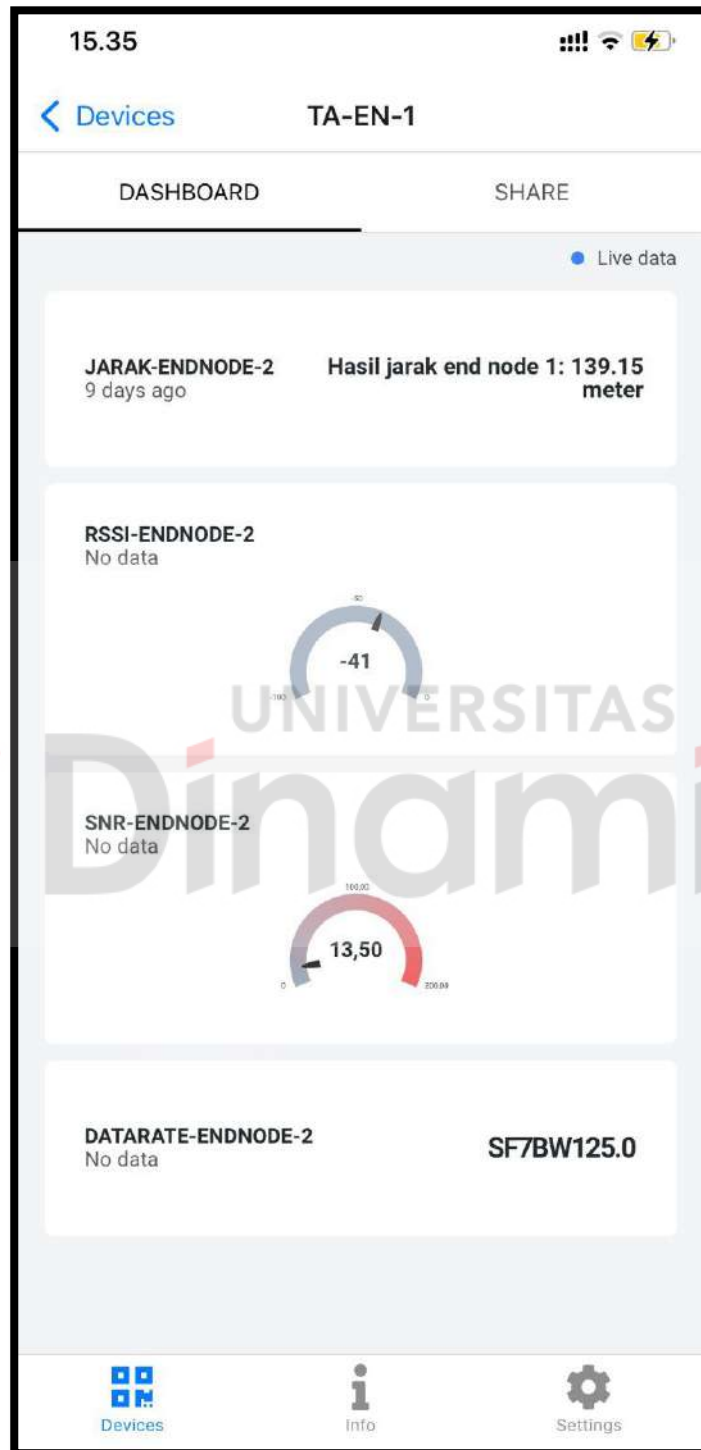
Gambar 3. 64 Title Device Datacake dari Aplikasi Handphone

Gambar 3. Menunjukkan perangkat *end node* yang telah berhasil ditambahkan, untuk melihat hasil visualisasi klik pada perangkat *end node*.



Gambar 3. 65 List Device Datacake Dari Aplikasi Handphone

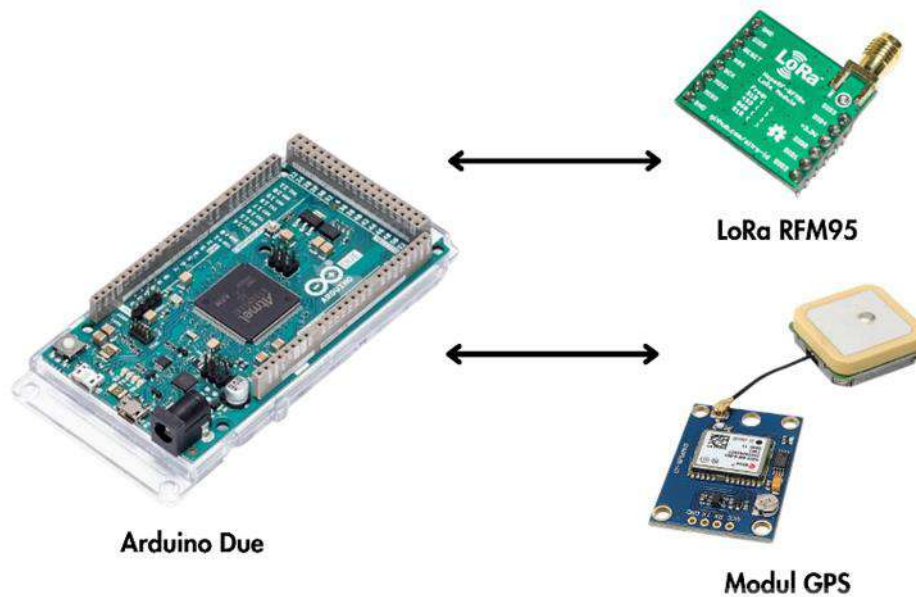
Gambar 3. Berhasil menampilkan dashboard visualisasi dari perangkat *end node* yang telah dibuat dari *datacake*.



Gambar 3. 66 Dashboard Device Datacake Dari Aplikasi Handphone

3.2 Perancangan Perangkat Keras

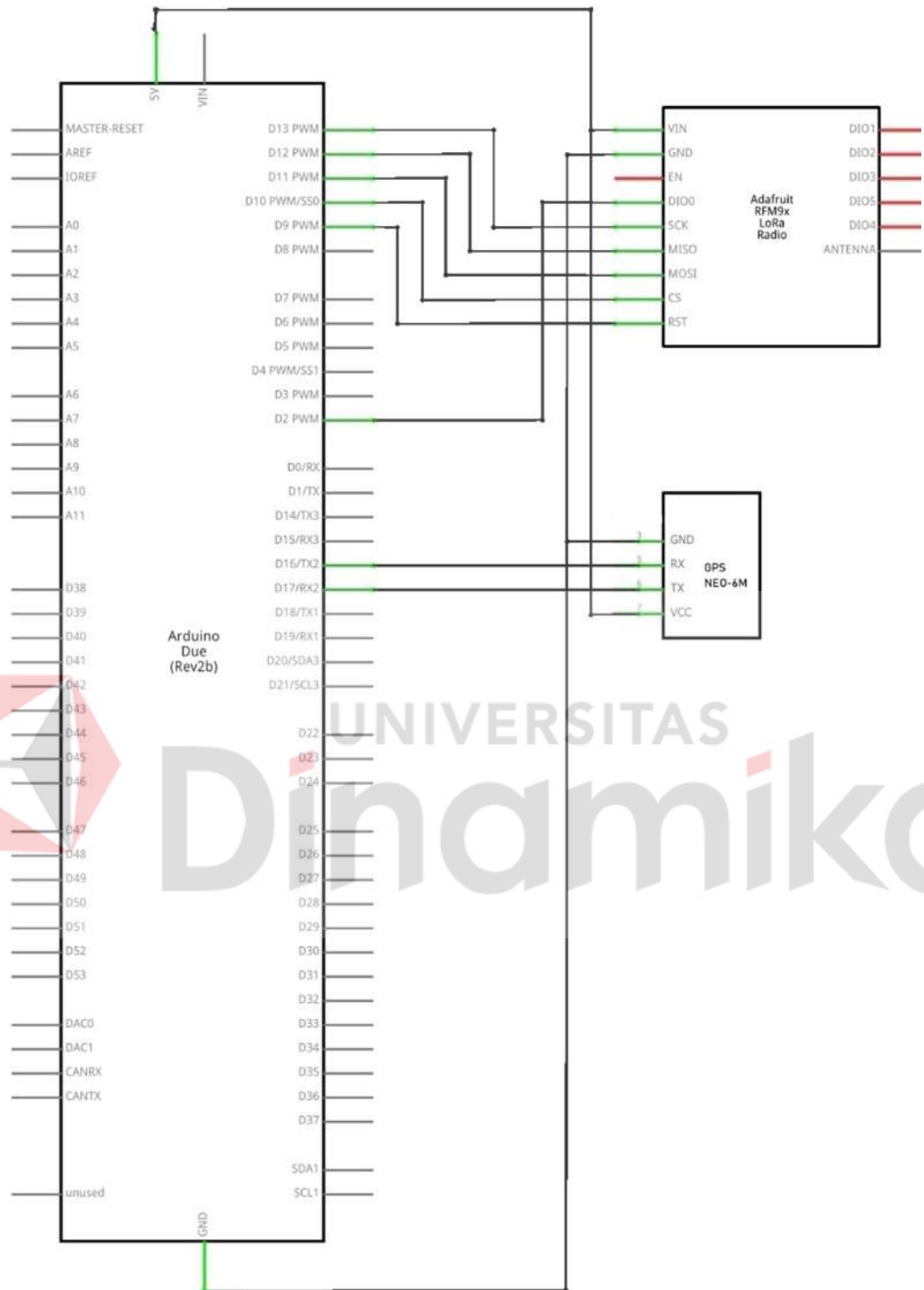
3.2.1 Perancangan LoRa *End Node*



Gambar 3. 67 Perancangan LoRa *End Node*

Perancangan perangkat LoRa *end node* menggunakan mikrokontroler Arduino yang dihubungkan dengan modul GPS dan modul LoRa *transmitter*. Jenis Arduino yang digunakan pada tugas akhir ini adalah Arduino Due. Sensor GPS yang digunakan adalah Neo 6M dan modul transmitter LoRa yang digunakan adalah RFM95 yang mendukung frekuensi 915MHz dan bisa dirubah menjadi 920-923MHz sesuai dengan peraturan yang telah ditentukan di Indonesia.

3.2.2 Rangkaian Skematik LoRa End Node



Gambar 3. 68 Skematik LoRa End Node

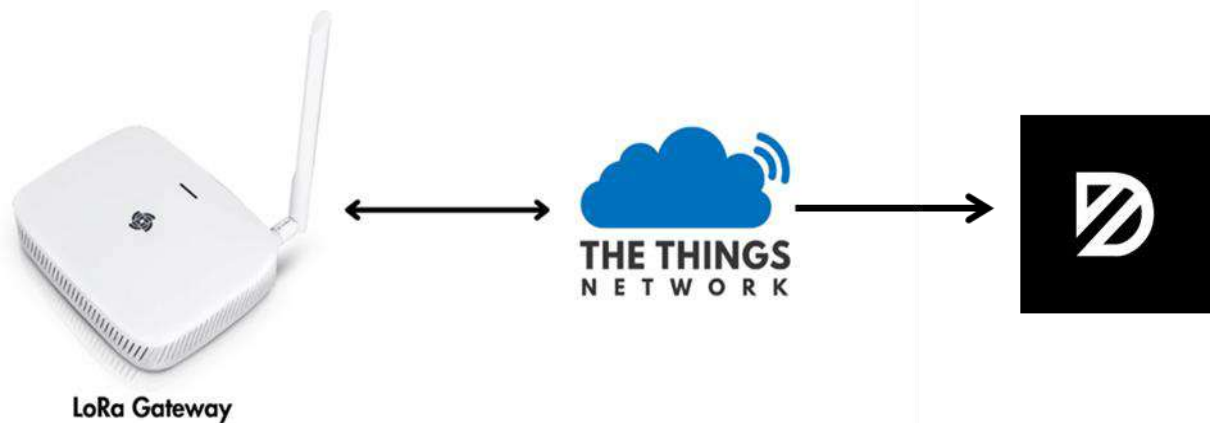
Gambar 3.68 menunjukkan rangkaian skematik yang digunakan untuk menghubungkan arduino due dengan modul transmitter LoRa dan sensor GPS. Komunikasi yang digunakan antara arduino dan modul LoRa adalah *Serial Peripheral Interface* (SPI) sedangkan Protokol Komunikasi yang digunakan antara

arduino dan sensor GPS adalah universal *asynchronous receiver / transmitter* (UART). Tabel 3.1 menunjukkan daftar PIN yang terhubung antara arduino ke modul transmitter LoRa dan sensor GPS.

Tabel 3. 1 PIN Arduino Due Ke *Transmitter* Lora dan Sensor GPS

Pin Arduino	Pin LoRa
Vcc	3.3V
Ground	Ground
D10	NSS
D5	Reset
MOSI	MOSI
MISO	MISO
SCK	SCK
D2	DIO0
D3	DIO1
Pin Arduino	Pin GPS
Vcc	5V
Ground	Ground
RX	TX
TX	RX

3.2.3 Perancangan Gateway



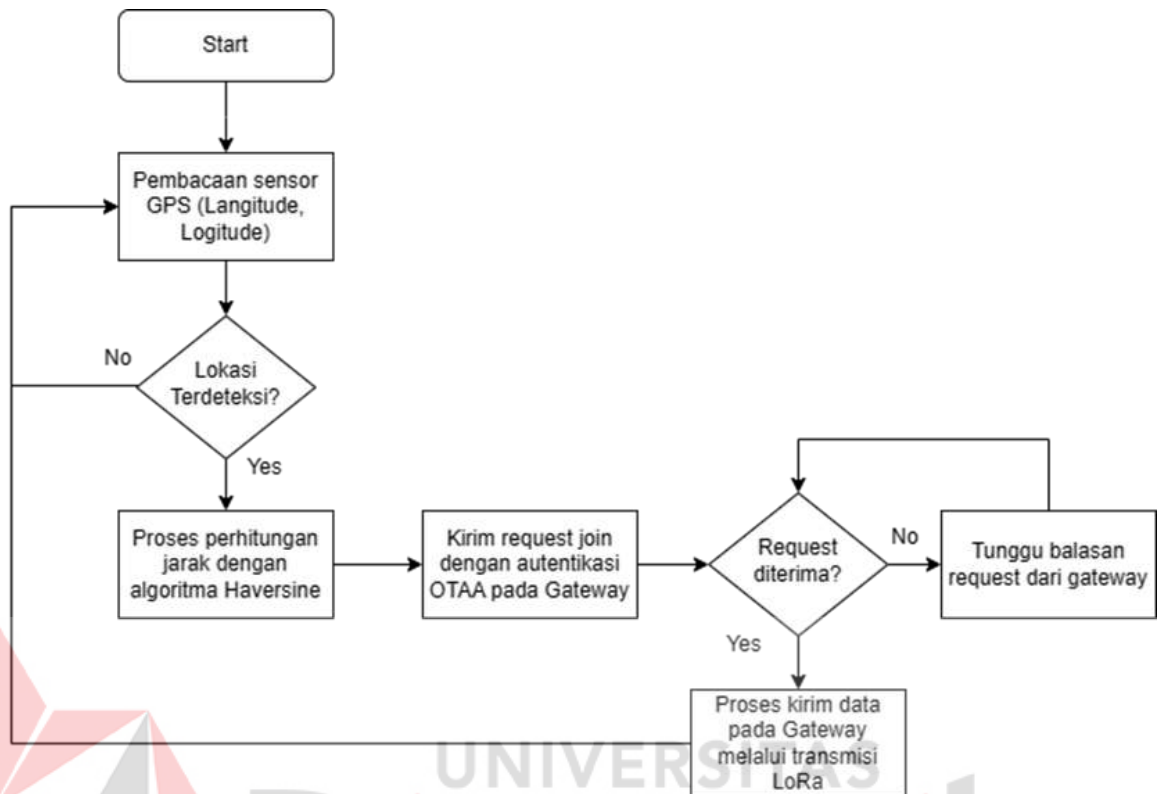
Gambar 3. 69 Perancangan Gateway ke Server TTN dan Datacake

Perangkat gateway yang digunakan pada Tugas Akhir ini adalah RAK7268. gateway RAK7268 support topologi LoRaWAN sehingga dapat terhubung secara langsung dengan server TNN dan meneruskan hasil data *haversine* ke Datacake melalui internet.

3.3 Perancangan Perangkat Lunak

Terdapat beberapa alur komunikasi data pada Tugas Akhir ini, mulai dari proses pembacaan posisi GPS hingga menampilkan data di server TTN dan Datacake. Berikut beberapa penerapan algoritma dari komunikasi data yang ada pada Tugas Akhir ini.

3.3.1 Algoritma Penerapan Sistem LoRa *End Node*

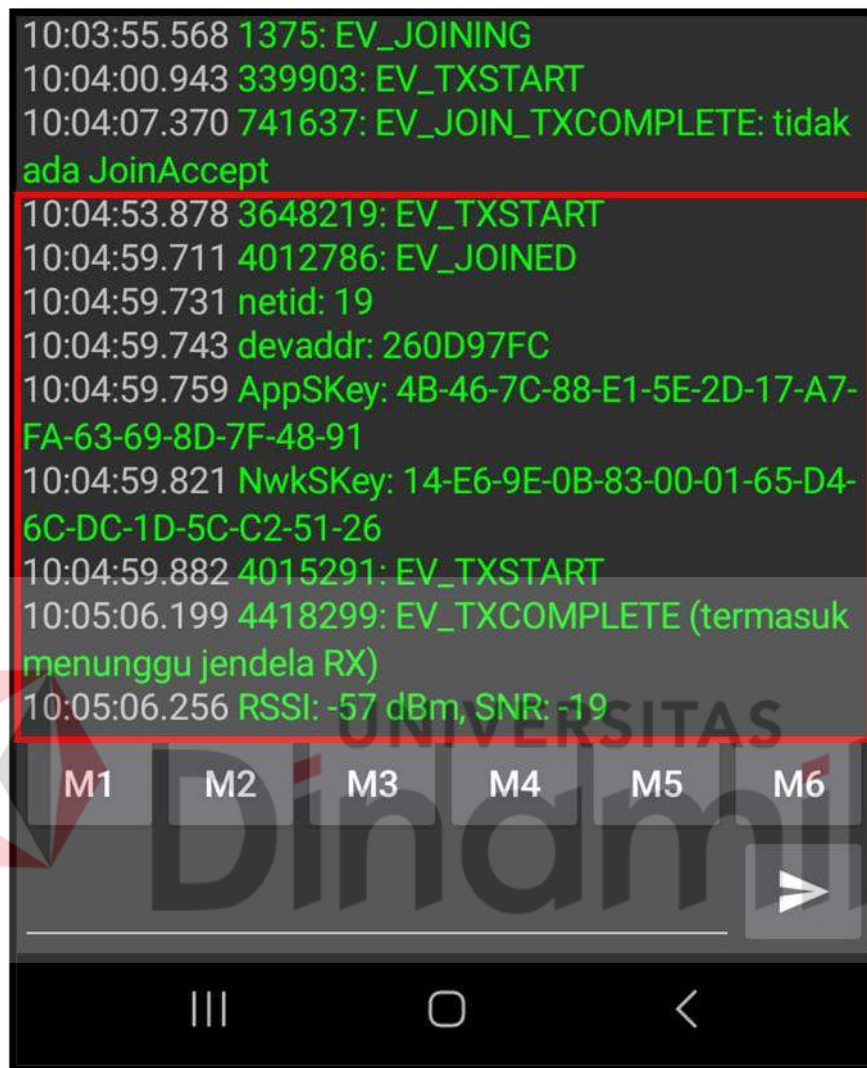


Gambar 3.70 Flowchart Sistem LoRa *End Node*

Proses algoritma sistem pada LoRa *end node* dimulai dari pembacaan data sensor GPS. Perhitungan jarak dengan metode *haversine* memerlukan data posisi dari *end node* dan juga *gateway*. Dikarenakan posisi dari *gateway* adalah tetap, maka data posisi dari *gateway* sudah diketahui terlebih dahulu. Sedangkan data posisi dari *end node* dapat berubah ubah (*mobile*) sehingga memerlukan data posisi dari sensor GPS. Setelah kedua data posisi dari *end node* maupun *gateway* sudah diketahui, perhitungan data jarak antara *end node* pada *gateway* menggunakan algoritma *haversine* dapat dilakukan.

Setelah perhitungan jarak telah selesai, langkah selanjutnya adalah mengirimkan data jarak menuju *gateway* dengan transmisi LoRa. Namun sebelum melakukan pengiriman data, *end node* harus terlebih dahulu melakukan autentikasi menggunakan *Over The Air Activation (OTAA)*. *End node* akan mengirimkan *request join* menuju *gateway* yang akan diteruskan pada server, setelah *end node* menerima balasan *request join* dari server maka *end node* bisa

mengirimkan data jarak menuju *gateway*. Gambar 3.71 Menunjukkan proses *request join* dari perangkat *end node* dan berhasil mendapatkan balasan dari server TTN.

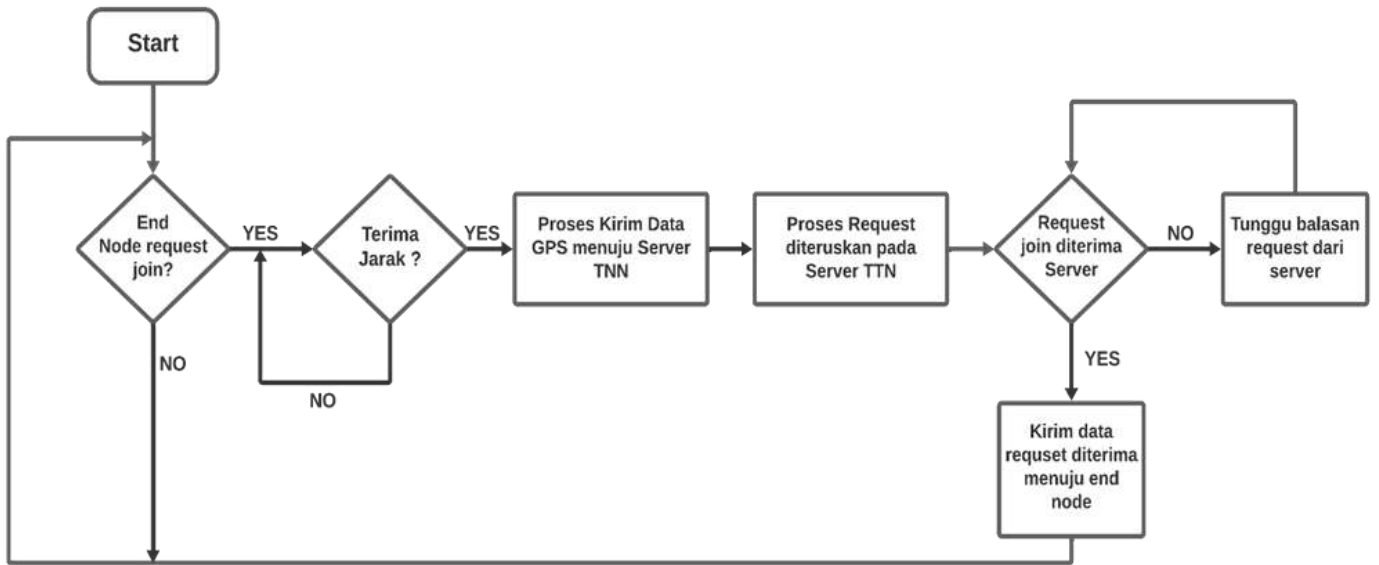


```
10:03:55.568 1375: EV_JOINING
10:04:00.943 339903: EV_TXSTART
10:04:07.370 741637: EV_JOIN_TXCOMPLETE: tidak
ada JoinAccept
10:04:53.878 3648219: EV_TXSTART
10:04:59.711 4012786: EV_JOINED
10:04:59.731 netid: 19
10:04:59.743 devaddr: 260D97FC
10:04:59.759 AppSKey: 4B-46-7C-88-E1-5E-2D-17-A7-
FA-63-69-8D-7F-48-91
10:04:59.821 NwkSKey: 14-E6-9E-0B-83-00-01-65-D4-
6C-DC-1D-5C-C2-51-26
10:04:59.882 4015291: EV_TXSTART
10:05:06.199 4418299: EV_TXCOMPLETE (termasuk
menunggu jendela RX)
10:05:06.256 RSSI: -57 dBm, SNR: -19
```

M1 M2 M3 M4 M5 M6

Gambar 3. 71 Tampilan *request join* Pada Serial monitor Handphone

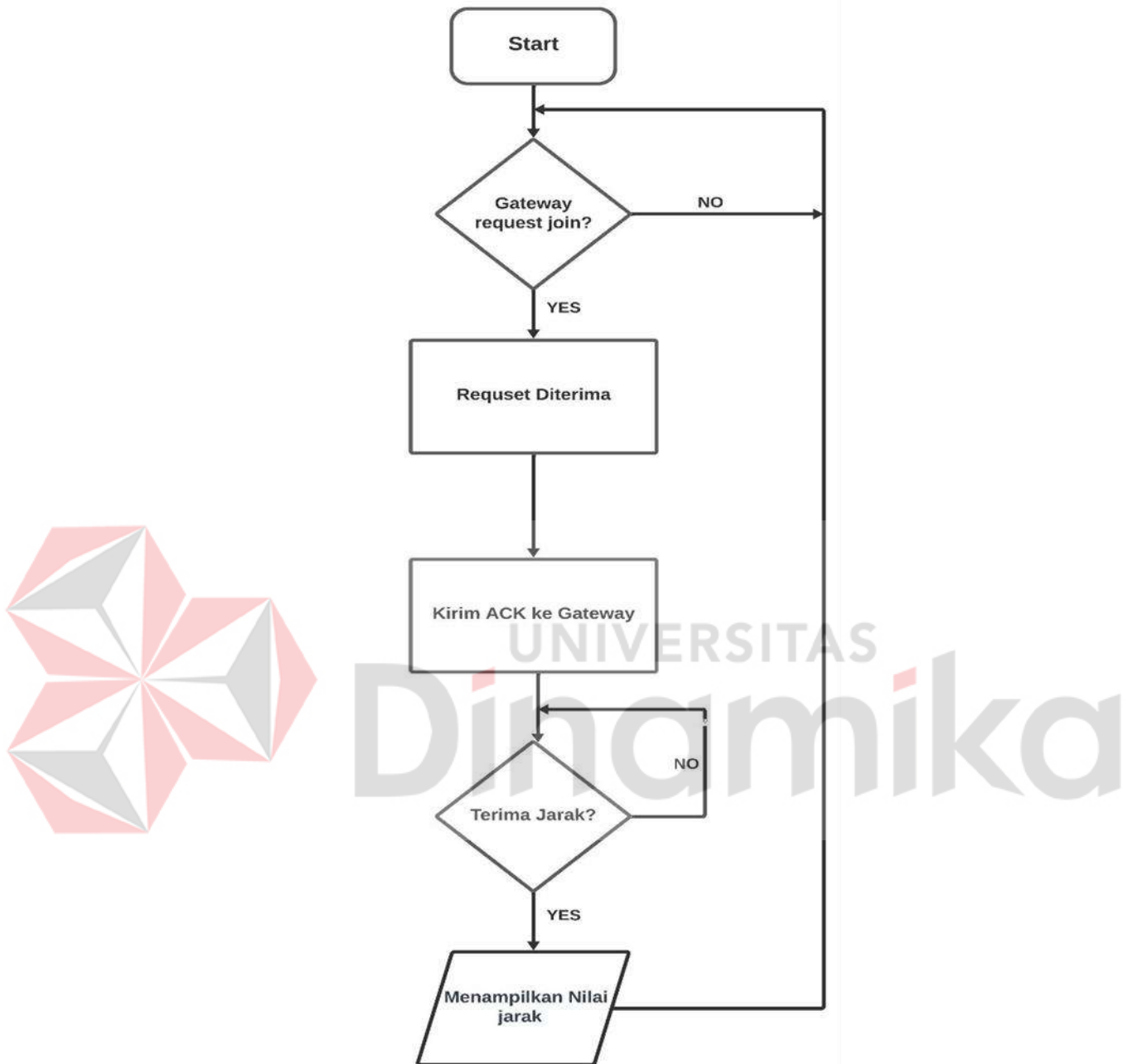
3.3.2 Algoritma Penerapan Sistem Pada Gateway



Gambar 3. 72 Flowchart Sistem Pada Gateway

Gambar 12 menunjukkan algoritma yang diterapkan pada gateway, dimana gateway memiliki tugas untuk menjadi jembatan dari end node dan juga server TTN. Terdapat dua tugas utama dari server, yaitu apabila gateway menerima request join yang dikirimkan oleh end node maka gateway akan melanjutkan request join menuju server sampai server merespon dengan balasan request diterima. Tugas kedua dari gateway adalah meneruskan data jarak yang dikirimkan oleh end node menuju server.

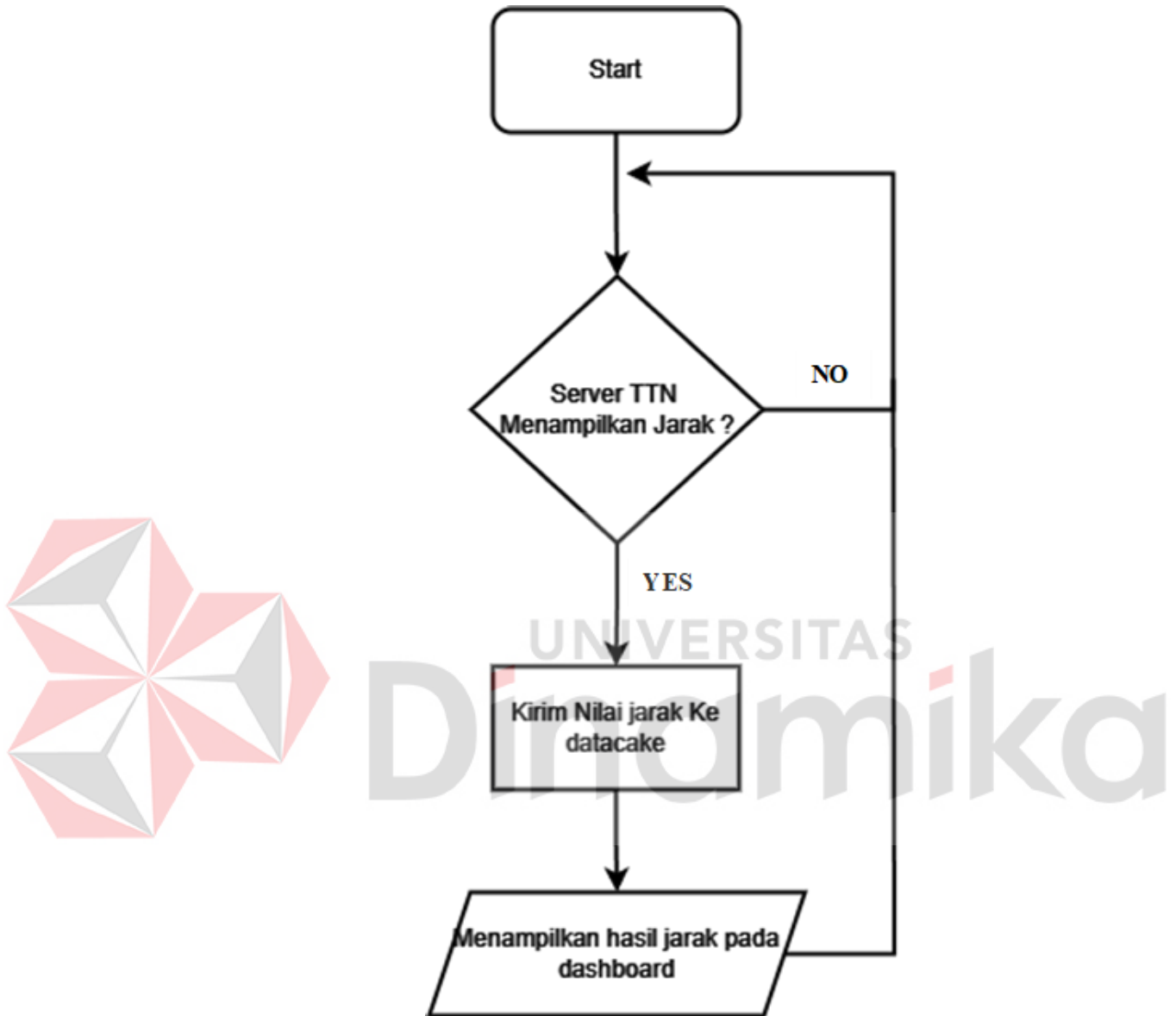
3.3.3 Algoritma Penerapan Sistem Pada Server TTN



Gambar 3. 73 Flowchart Sistem Pada Server TTN

Proses pengiriman yang dilakukan oleh sistem akan masuk pada server TTN. Selain melakukan penerimaan data yang dikirimkan oleh *gateway*, server juga bertugas membalas semua *request join* yang telah dikirimkan oleh *end node* melalui *gateway*, dan meneruskan data yang diterima ke *datacake*.

3.3.4 Algoritma Penerapan Sistem Pada *Datacake*



Gambar 3. 73 Flowchart Sistem Pada *Datacake*

Saat server TTN berhasil menerima dan menampilkan data jarak, maka akan mengirimkannya ke *Datacake* lalu di visualisasikan agar lebih rapih dan menarik.

BAB IV

HASIL DAN PEMBAHASAN

Bab 4 berisi hasil dari beberapa pengujian yang dilakukan serta pembahasan detail mengenai hasil pengujian. Dibawah ini akan dijelaskan hasil dari tiga pengujian yang telah dilakukan:

4.1 Pengujian Pengiriman Data *End Node* pada *Gateway* dan Server TTN

4.1.1 Tujuan

Tujuan dari pengujian pengiriman data dari *end node* pada *gateway* adalah:

1. Mengetahui pola dari nilai RSSI yang diperoleh pada area pengujian,
2. Memastikan data jarak yang telah diolah menggunakan metode *haversine* dapat terkirim pada perangkat *gateway* dan server TTN,
3. Membandingkan data jarak dari hasil perhitungan dengan metode *haversine* dengan data jarak sebenarnya.

4.1.2 Peralatan yang Digunakan

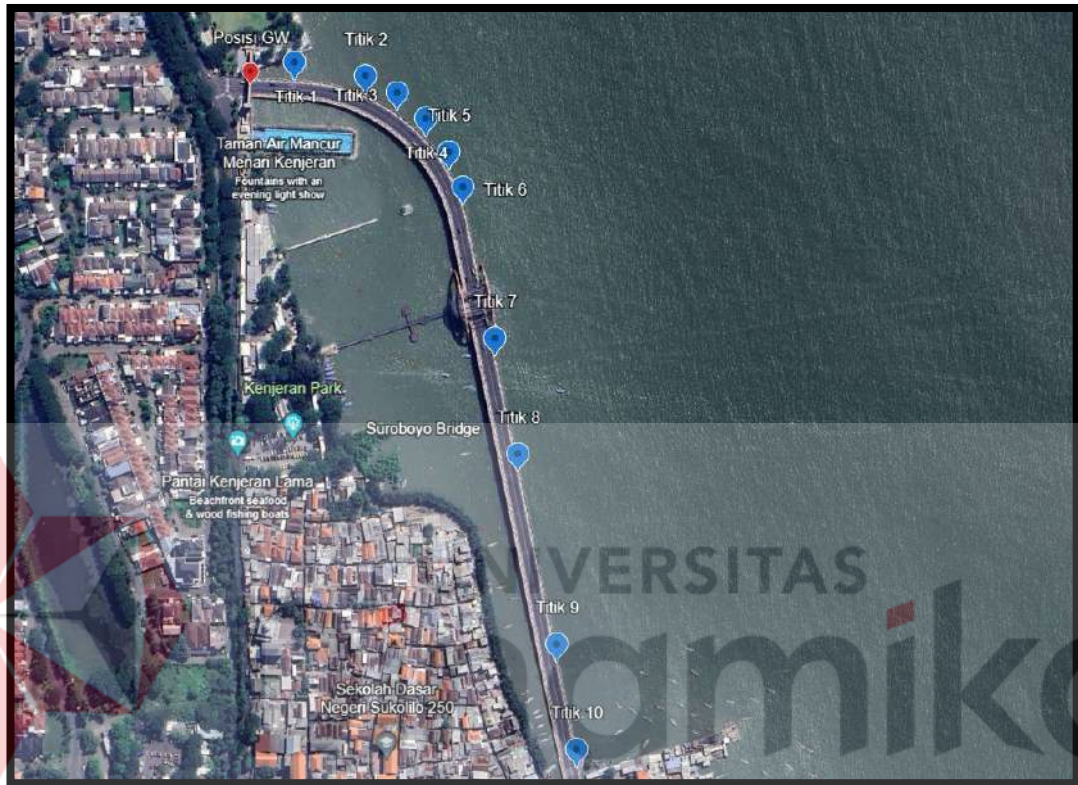
1. 1 perangkat *end node* (Arduino, modul GPS, dan LoRa).
2. 1 modul *gateway* sebagai perantara antara *end node* dan server TTN.
3. 1 Laptop untuk pengambilan data pada *receiver*.

4.1.3 Cara Pengujian

1. *Mengupload* program perhitungan jarak dan pengiriman data dengan transmisi LoRa yang telah dibuat pada mikrokontroler Arduino.
2. Melakukan pengujian pada tiap jarak yang sudah ditentukan.
3. Menyimpan hasil pengiriman data yang ada pada server TTN.
4. Mengamati hasil dari pengujian yang telah dilakukan.

4.1.4 Skenario Pengujian Pertama

Pengujian ini dilakukan dengan mengirimkan data jarak secara *point to point* menggunakan transmisi LoRa. Pengujian dilakukan pada area pesisir pantai Kenjeran kota Surabaya. Gambar 4.1 menunjukkan lokasi pengujian yang akan dilakukan.



Gambar 4. 1 Lokasi Skenario Pengujian Pertama

Terdapat 10 titik pengujian pengiriman data dari *end node* pada perangkat *gateway*. Pengujian ini dilakukan dengan melakukan pengiriman data dengan jarak yang berbeda beda, jarak pengujian yang dilakukan dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Jarak *Google Earth* Pengujian Pertama

Titik	Jarak (Google Earth)
1	36,44 meter
2	95,30 meter
3	123,14 meter
4	150,73 meter
5	178,71 meter

6	202,26 meter
7	301,40 meter
8	405,32 meter
9	536,74 meter
10	621,98 meter

Pada pengujian ini akan dilakukan perbandingan data jarak aktual area pesisir pantai kenjeran kota Surabaya dengan data dari perhitungan *haversine* yang diperoleh dari koordinat sensor GPS. Sensor GPS pada perangkat *end node* menghasilkan koordinat *latitude* dan *longitude*. Untuk melakukan perhitungan jarak menggunakan metode *haversine*, memerlukan data koordinat *latitude* dan *longitude* dari posisi pengirim dan posisi penerima. Sensor GPS pada perangkat *end node* hanya melakukan pengambilan data koordinat dari sisi pengirim, sedangkan penerima koordinatnya sudah ditentukan dalam program *end node* dikarenakan posisi *gateway* tidak berpindah tempat atau tetap. Persamaan yang digunakan dapat dilihat pada landasan teori untuk menghitung jarak dengan perhitungan *haversine*.



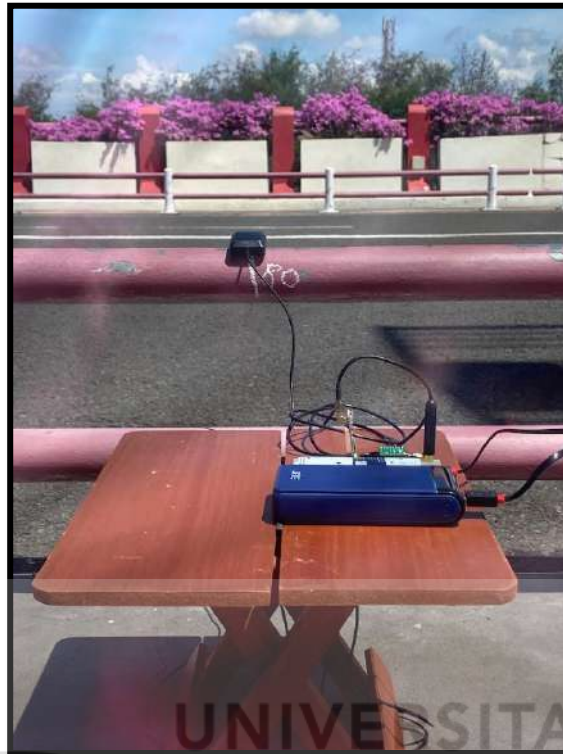
UNIVERSITAS
Dinamika

Gambar 4.2 menunjukkan posisi *gateway* pada pengujian pertama yang akan dilakukan.



Gambar 4. 2 Posisi *Gateway* Skenario *Pengujian Pertama*

Gambar 4.3 Menunjukkan posisi perangkat *end node* pada pengujian pertama yang akan dilakukan.

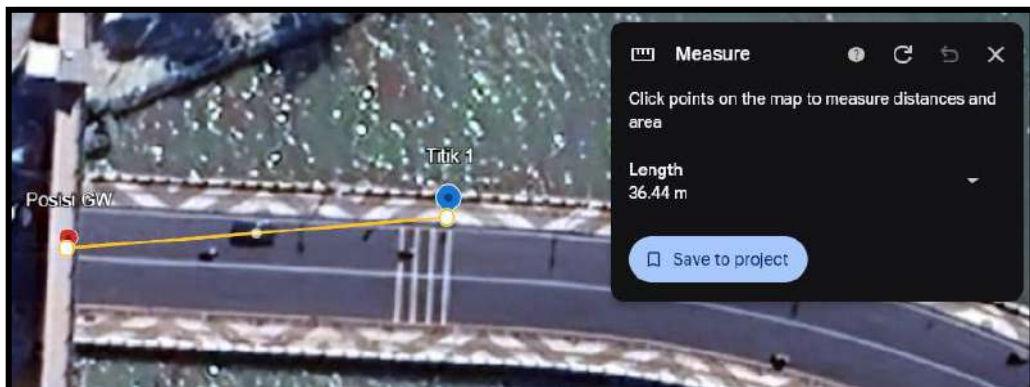


Gambar 4. 3 Posisi Perangkat *End Node*

4.1.5 Hasil Pengujian

Pengujian dilakukan dengan pengambilan data yang diterima oleh *gateway* dari masing-masing posisi yang sudah ditentukan, berikut penjelasan hasil dari data pengujian yang telah diperoleh dari posisi 1 sampai 10.

1. Posisi 1 (36,44m)



Gambar 4. 4 Pengukuran Jarak Posisi 1 *Google Earth*

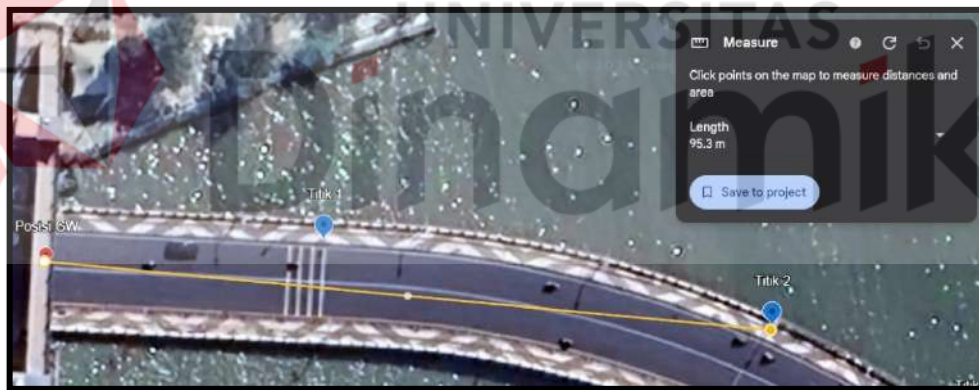
Tabel 4.2 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi pertama.

Tabel 4. 2 Hasil Pengujian *end node* Posisi 1

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
37,77	-51	13,25	7	0,06
38,43	-50	13,25	7	0,06
38,18	-53	13,75	8	0,10

Hasil pengujian pada posisi pertama menghasilkan rata-rata jarak sebesar 38,13 meter, dengan rata-rata RSSI sebesar -51,33, rata-rata SNR sebesar 13,45, dan nilai SF 7 sampai 8 sehingga memperlambat waktu dari parameter *Airtime* dengan rentang 0,05s sampai 0,10s.

2. Posisi 2 (95,3m)



Gambar 4. 5 Pengukuran Jarak Posisi 2 *Google Earth*

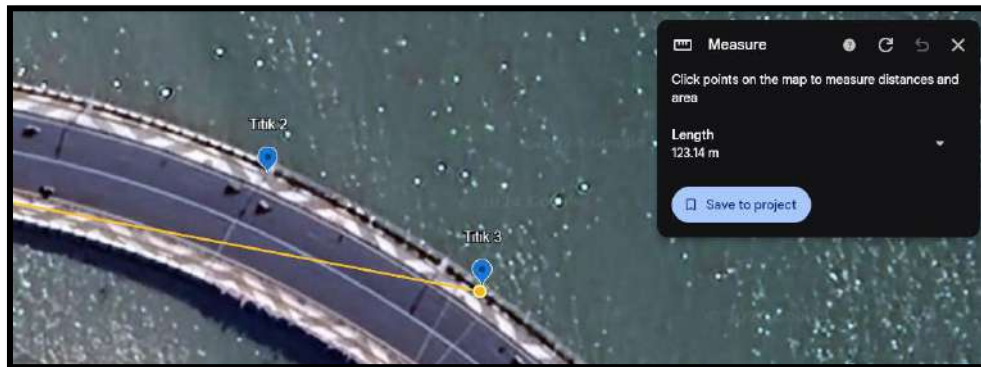
Tabel 4.3 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi kedua.

Tabel 4. 3 Hasil Pengujian *end node* Posisi 2

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
97,35	-65	10,50	7	0,05
97,53	-67	10,75	7	0,06
99,69	-56	13,50	7	0,06

Hasil pengujian pada posisi kedua menghasilkan rata-rata jarak sebesar 98,63 meter, dengan rata-rata RSSI sebesar -59,20, rata-rata SNR sebesar 11,90, dan nilai SF 7 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,06s sampai 0,05s.

3. Posisi 3 (123,14m)



Gambar 4. 6 Pengukuran Jarak Posisi 3 *Google Earth*

Tabel 4.4 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi ketiga.

Tabel 4. 4 Hasil Pengujian *end node* Posisi 3

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
124,22	-61	10.25	7	0,06
125,29	-62	12.75	8	0,10
125,39	-62	7.75	8	0,10

Hasil pengujian pada posisi ketiga menghasilkan rata-rata jarak sebesar 125,11 meter, dengan rata-rata RSSI sebesar -64,40, rata-rata SNR sebesar 9,75, dan nilai SF 7 sampai 8 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,06s sampai 0,10s.

4. Posisi 4 (150,73m)



Gambar 4. 7 Pengukuran Jarak Posisi 4 *Google Earth*

Tabel 4.5 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi keempat.

Tabel 4. 5 Hasil Pengujian *end node* Posisi 4

Jarak (m)	RSSI	SNR	SF	Airtime (s)
152,74	-72	8,25	7	0,06
154,23	-62	12	8	0,10
153,76	-61	9,25	8	0,10

Hasil pengujian pada posisi keempat menghasilkan rata-rata jarak sebesar 153,63 meter, dengan rata-rata RSSI sebesar -64,40, dan rata-rata SNR sebesar 9,25, dan nilai SF 7 sampai 8 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,06s sampai 0,10s.

5. Posisi 5 (178,71m)



Gambar 4. 8 Pengukuran Jarak Posisi 5 *Google Earth*

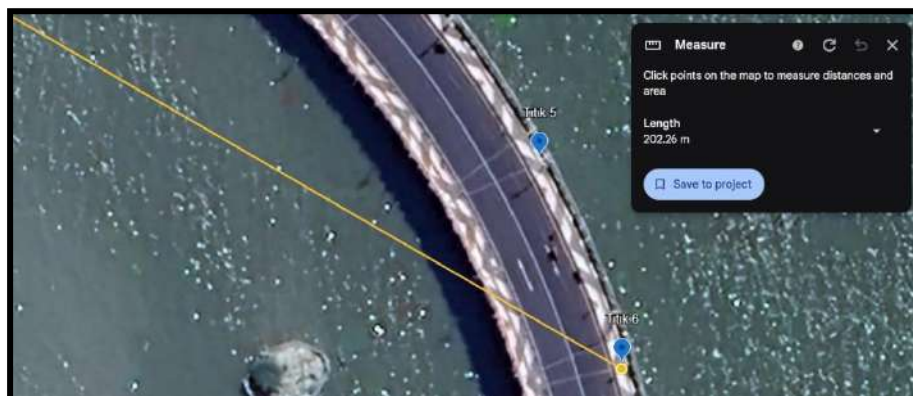
Tabel 4.6 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi kelima.

Tabel 4. 6 Hasil Pengujian *end node* Posisi 5

Jarak (m)	RSSI	SNR	SF	Airtime (s)
179,65	-71	5,75	8	0,10
179,61	-70	7,25	8	0,10
179,56	-70	8,25	8	0,09

Hasil pengujian pada posisi kelima menghasilkan rata-rata jarak sebesar 179,61 meter, dengan rata-rata RSSI sebesar -69,50, rata-rata SNR sebesar 7,69, dan nilai SF 8 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,09s sampai 0,10s.

6. Posisi 6 (202,26m)



Gambar 4. 9 Pengukuran Jarak Posisi 6 *Google Earth*

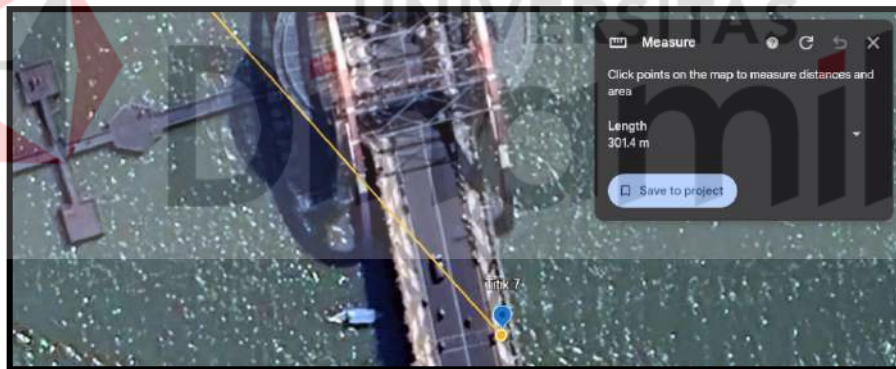
Tabel 4.7 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi keenam.

Tabel 4. 7 Hasil Pengujian *end node* Posisi 6

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
206,31	-79	9	8	0,10
206,03	-75	8	9	0,21
206,89	-83	5	10	0,37

Hasil pengujian pada posisi keenam menghasilkan rata-rata jarak sebesar 206,35 meter, dengan rata-rata RSSI sebesar -78,67, rata-rata SNR sebesar 7,33, dan nilai SF 8 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,10s sampai 0,37s.

7. Posisi 7 (301,4m)



Gambar 4. 10 Pengukuran Jarak Posisi 7 *Google Earth*

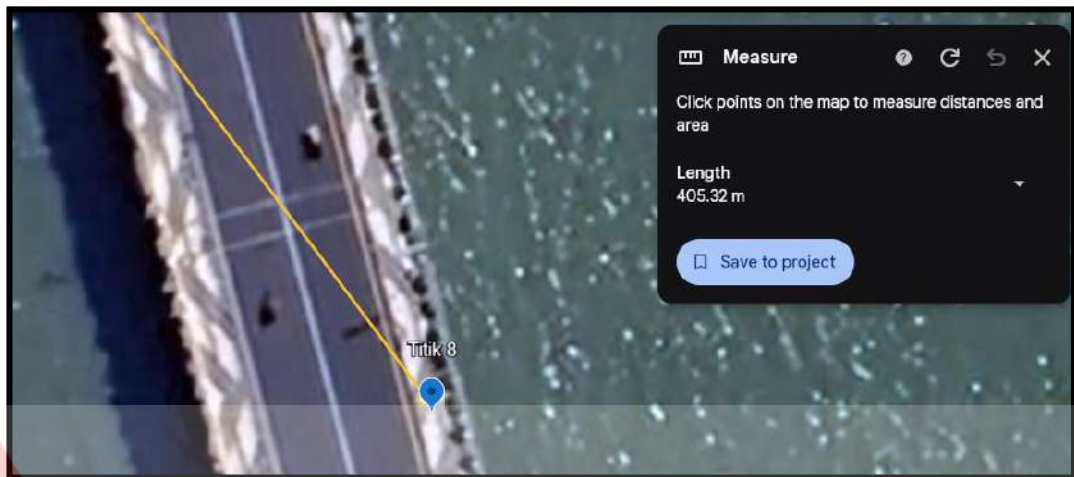
Tabel 4.8 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi ketujuh.

Tabel 4. 8 Hasil Pengujian *end node* Posisi 7

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
308,60	-91	0,50	10	0,37
306,45	-91	1,75	10	0,37
307,52	-92	1	10	0,37

Hasil pengujian pada posisi ketujuh menghasilkan rata-rata jarak sebesar 307,53 meter, dengan rata-rata RSSI sebesar -91,33, rata-rata SNR sebesar 1,08, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,37s.

8. Posisi 8 (405,32m)



Gambar 4. 11 Pengukuran Jarak Posisi 8 *Google Earth*

Tabel 4.9 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi kedelapan.

Tabel 4. 9 Hasil Pengujian *end node* Posisi 8

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
394,29	-84	4,25	10	0,37
396,29	-85	3,75	10	0,37
394,24	-82	4,75	10	0,37

Hasil pengujian pada posisi kedelapan menghasilkan rata-rata jarak sebesar 394,94 meter, dengan rata-rata RSSI sebesar -83,67, dan rata-rata SNR sebesar 4,25, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,37s.

9. Posisi 9 (536,74m)



Gambar 4. 12 Pengukuran Jarak Posisi 9 *Google Earth*

Tabel 4.10 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi kesembilan.

Tabel 4. 10 Hasil Pengujian *end node* Posisi 9

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
543,33	-100	-2,75	10	0,37
543,80	-97	-0,75	10	0,37
543,56	-98	-2,50	10	0,37

Hasil pengujian pada posisi kesembilan menghasilkan rata-rata jarak sebesar 543,57 meter, dengan rata-rata RSSI sebesar -98,33, dan rata-rata SNR sebesar -2,00, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,37s.

10. Posisi 10 (621,98m)



Gambar 4. 13 Pengukuran Jarak Posisi 10 *Google Earth*

Tabel 4.11 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi kesepuluh.

Tabel 4. 11 Hasil Pengujian *end node* Posisi 10

Jarak (m)	RSSI	SNR	SF	Airtime (s)
639,06	-106	-11,25	10	0,37
639,11	-107	-13,50	10	0,37
639,63	-104	-5,75	10	0,37

Hasil pengujian pada posisi kesepuluh menghasilkan rata-rata jarak sebesar 639,27 meter, dengan rata-rata RSSI sebesar -105,67, dan rata-rata dari SNR sebesar -10,17, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,37s.

4.1.6 Analisis Data

Tabel 4.12 menunjukkan hasil rangkuman dari data jarak yang diperoleh dari 10 titik percobaan yang telah dilakukan.

Tabel 4. 12 Hasil Analisis Pengujian Skenario Pertama

Titik	Rata-Rata RSSI (dBm)	Rata-Rata SNR (dBm)	Airtime (s)	SF	Haversine (meter)	Selisih Jarak (meter)	Persentase Error %
1	-51,33	13,42	0,07	7	38,13	1,69	4,64%
2	-59,20	11,90	0,12	7	98,63	3,33	3,52%
3	-64,40	9,75	0,14	8	125,11	1,97	1,60%
4	-64,40	9,25	0,15	8	153,63	2,90	1,93%
5	-69,50	7,69	0,10	8	179,61	0,90	0,50%
6	-78,67	7,33	0,20	8	206,35	4,09	2,02%
7	-91,33	1,08	0,37	10	307,53	6,13	2,03%
8	-83,67	4,25	0,37	10	394,94	10,37	2,56%
9	-98,33	-2,00	0,37	10	543,57	6,82	1,27%
10	-105,67	-10,17	0,37	10	639,27	17,29	2,78%

Dilihat dari hasil pengujian yang dilakukan, sinyal RSSI yang dihasilkan mempunyai perubahan sinyal dengan rentang -51,33 hingga -105,67. Besar sinyal RSSI pada pengujian ini mengikuti dari jarak dari proses pengujian, hanya ada satu titik yang mengalami penurunan RSSI yaitu pada pengujian pada titik 8, secara keseluruhan data diatas membuktikan semakin jauh jarak pengujian maka semakin menurun kualitas sinyal yang diterima atau nilai RSSI yang diperoleh semakin kecil.

Nilai SNR yang dihasilkan mempunyai perubahan sinyal yang baik dengan rentang 13,42 hingga -10,17, semakin besar nilai SNR maka semakin baik komunikasi yang dilakukan, berdasarkan data diatas semakin jauh titik dari *end node* maka semakin kecil nilai SNR yang dihasilkan, sehingga menyebabkan banyak *noise* sinyal yang dihasilkan.

SF yang dihasilkan mempunyai perubahan nilai dari 7 hingga 10, semakin tinggi nilai SF maka akan memperlambat laju data tetapi memungkinkan komunikasi yang lebih andal dalam kondisi sinyal yang buruk atau pada jarak yang lebih jauh.

Nilai *Airtime* menunjukkan berapa lama data dapat terkirim melalui transmisi LoRa, berdasarkan data diatas menghasilkan rentang nilai *Airtime* 0,07s sampai 0,37s, semakin jauh posisi dari *end node* maka semakin besar nilai *airtime* yang dihasilkan. Sedangkan konversi jarak menggunakan metode *haversine* dapat dikatakan cukup akurat dengan nilai presentasi *error* di bawah 5%.

4.2 Skenario Pengujian Kedua

4.2.1 Tujuan

Pengujian komunikasi *multi node* dilakukan untuk memastikan seluruh data dari tiga *end node* dapat di terima secara *real-time* bersama-sama serta menguji performansi komunikasi lora di atas laut. Memastikan data yang dikirimkan dari perangkat *end node* LoRa dapat dikirim menuju server TTN melalui *gateway* dan kemudian ditampilkan secara grafik pada aplikasi *datacake*.

4.2.2 Peralatan yang Digunakan

1. 3 perangkat *end node* (Arduino, modul GPS, dan LoRa).
2. 1 modul *gateway* sebagai perantara antara *end node* dan server TTN.
3. 1 Laptop untuk pengambilan data pada *receiver*.

4.2.3 Cara Pengujian

1. *Mengupload* program pengiriman data jarak *haversine* dengan transmisi LoRa yang telah dibuat pada mikrokontroller Arduino.
2. Melakukan pengujian pada tiap titik yang sudah ditentukan.
3. Menyimpan hasil pengiriman data yang ada pada server TTN.
4. Mengamati hasil dari pengujian yang telah dilakukan.

4.2.4 Pengujian Komunikasi *Multi Node*



Gambar 4. 14 Skenario Pengujian Kedua

Pada Gambar 4.14 menunjukkan skenario pengujian keseluruhan sistem, dalam skenario pengujian ini LoRa gateway terhubung pada area yang tersedia jaringan WiFi agar dapat terhubung secara langsung dengan server TNN melalui internet. Sedangkan untuk posisi masing-masing perangkat *end node 1*, *end node 2*, dan *end node 3* berada diluar dari area jangkauan WiFi dengan pengujian jarak yang akan ditentukan. Masing-masing perangkat *end node* akan mengirimkan data jarak yang telah diolah menggunakan metode *haversine* untuk mengetahui seberapa jauh jarak dari perangkat *end node* terhadap posisi LoRa gateway dan menganalisis seberapa baik komunikasi perangkat *end node* dengan LoRa gateway di daerah pesisir Pantai.

Lokasi dari *end node* 1 dan *end node* 3 akan bersifat tetap, sedangkan posisi dari *end node* 2 akan berpindah pindah (semakin jauh).



Gambar 4. 15 Skenario Pengujian Kedua Perangkat *end node* 2

Posisi *end node* 1 dan *end node* 3 adalah tetap, Tabel 4.13 menampilkan jarak pengujian yang akan dilakukan.

Tabel 4. 13 Jarak *Google Earth* Pengujian Kedua

End Node	Jarak (Google earth)
1	104,86 meter
3	140,69 meter

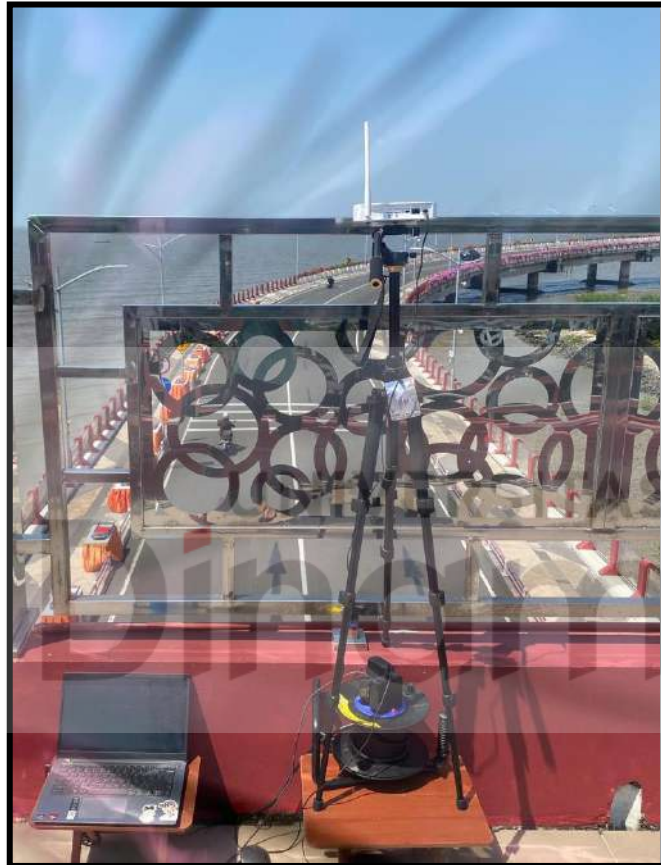
Terdapat 5 titik pengujian pengiriman data dari *end node* 2. Pengujian ini dilakukan dengan melakukan pengiriman data dengan jarak yang berbeda beda, Tabel 4.14 menampilkan jarak pengujian yang akan dilakukan.

Tabel 4. 14 Jarak GPS Garmin Pengujian Kedua

Titik	Jarak (Garmin)
1	238 meter
2	427 meter

3	604 meter
4	800 meter
5	1020 meter

Gambar 4.16 menunjukkan posisi *gateway* pada pengujian kedua yang akan dilakukan.



Gambar 4. 16 Posisi *Gateway Skenario Pengujian Kedua*

Gambar 4.17 menunjukkan posisi perangkat *end node 1* pada skenario pengujian kedua yang akan dilakukan.



Gambar 4. 17 Posisi Perangkat *End Node 1*

Gambar 4.18 menunjukkan posisi perangkat *end node 2* pada pengujian kedua yang berada di atas perahu.



Gambar 4. 18 Posisi Perangkat *End Node 2*

Gambar 4.18 menunjukkan posisi perangkat *end node* 3 pada pengujian kedua yang akan dilakukan.

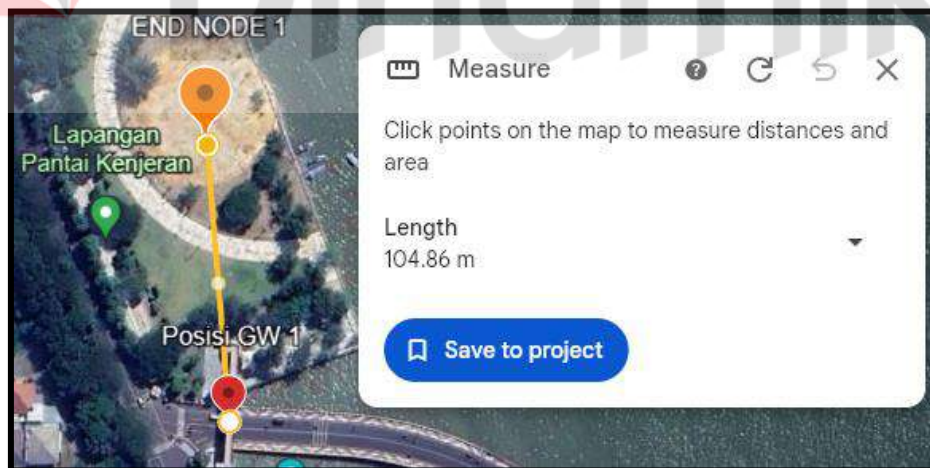


Gambar 4. 19 Posisi Perangkat *End Node* 3

4.2.5 Hasil Pengujian

Pengujian seluruh sistem dibagi menjadi 5 sesi pengambilan data, dimana setiap *end node* melakukan pengujian sebanyak 5 kali. Dibawah ini adalah hasil dari pengujian tiap *end node*.

1. Pengujian *end node* 1



Gambar 4. 20 Pengukuran Jarak *end node* 1 dengan *Google Earth*

Tabel 4.15 menampilkan hasil dari pengujian *end node* 1, yang dimulai dari sesi 1 sampai 5.

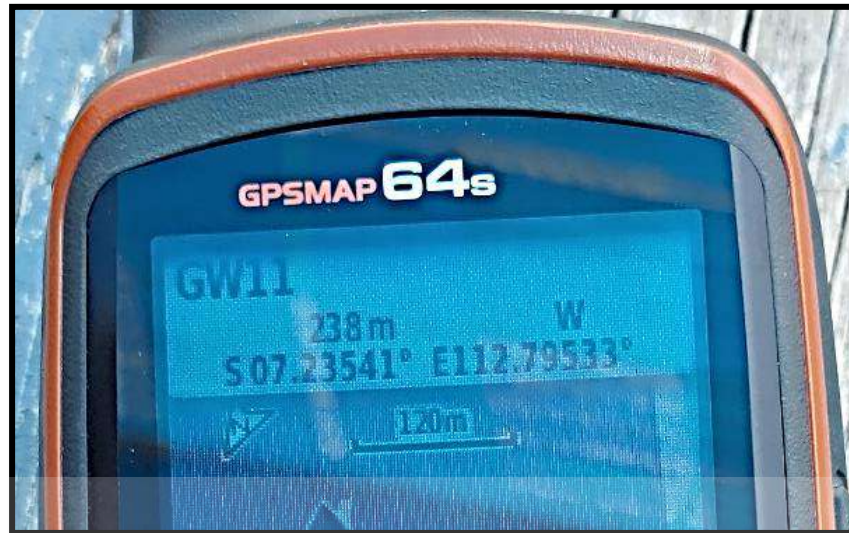
Tabel 4. 15 Hasil Pengujian *end node* 1 Skenario 2

Sesi	Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
1	104,98	-81	5,75	9	0,19
	104,87	-80	6,25	9	0,19
	103,94	-80	3,50	10	0,33
2	103,80	-85	4	9	0,19
	104,55	-89	1,75	9	0,19
	105,38	-90	3	9	0,19
3	103,76	-89	2,5	9	0,19
	103,76	-87	3,5	9	0,19
	104,13	-86	3,5	9	0,19
4	105,32	-84	3,75	8	0,10
	105,18	-84	6,25	8	0,10
	105,18	-84	5,5	8	0,10
5	103,82	-83	6,25	7	0,05
	104,14	-82	7	7	0,05
	104,14	-82	6,75	7	0,05

Hasil dari pengujian pada *end node* 1 menghasilkan rata-rata jarak sebesar 104,62 meter, dengan rata-rata RSSI sebesar -85,16, rata-rata SNR sebesar 4,06, dan nilai SF 7 sampai 10 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,05s sampai 0,37s.

1. Pengujian *end node 2*

Gambar 4.21 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin, alat ini digunakan sebagai pembanding dengan jarak yang dikirimkan oleh *end node* menggunakan metode *haversine*.



Gambar 4. 21 Jarak *end node 2* Sesi 1 dengan GPS Garmin

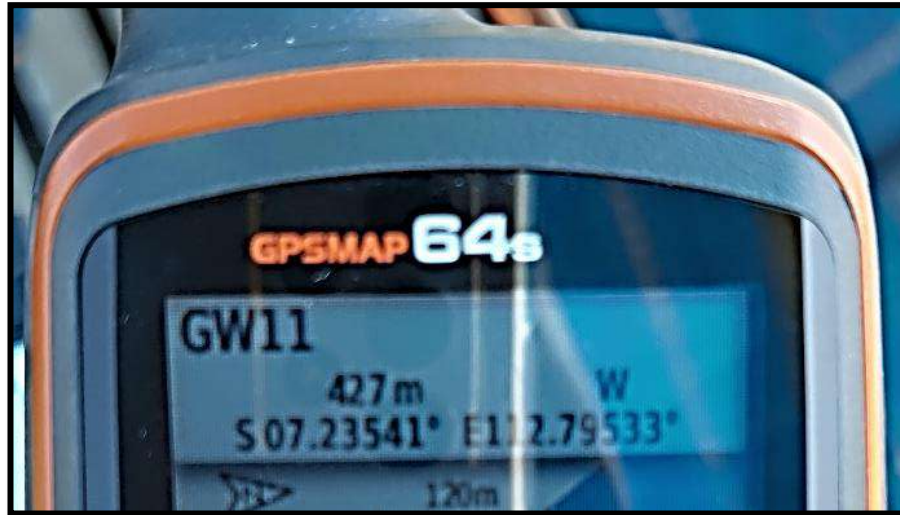
Tabel 4.16 menampilkan hasil pengujian dari perangkat *end node 2* menggunakan metode *haversine* pada sesi pertama.

Tabel 4. 16 Hasil Pengujian *end node 2* sesi 1

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
236,97	-85	3,25	10	0,33
237,27	-82	4,50	10	0,33
237,27	-83	4,00	10	0,33

Hasil dari pengujian pada *end node 2* sesi pertama menghasilkan rata-rata jarak sebesar 241,28 meter, dengan rata-rata RSSI sebesar -88,55, rata-rata SNR sebesar 0,22, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,29s sampai 0,37s.

Gambar 4.22 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada *end node 2* sesi kedua.



Gambar 4. 22 Jarak *end node 2* Sesi 2 dengan GPS Garmin

Tabel 4.17 menampilkan hasil pengujian dari perangkat *end node 2* menggunakan metode *haversine* pada sesi kedua.

Tabel 4. 17 Hasil Pengujian *end node 2* sesi 2

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
428,32	-89	1,75	10	0,33
428,53	-95	-2	10	0,33
428,75	-92	-0,5	10	0,33

Hasil dari pengujian pada *end node 2* sesi kedua menghasilkan rata-rata jarak sebesar 213,42 meter, dengan rata-rata RSSI sebesar -88,89, rata-rata SNR sebesar -1,46, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,33s.

Gambar 4.23 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada *end node 2* sesi tiga.



Gambar 4. 23 Jarak *end node 2* Sesi 3 dengan GPS Garmin

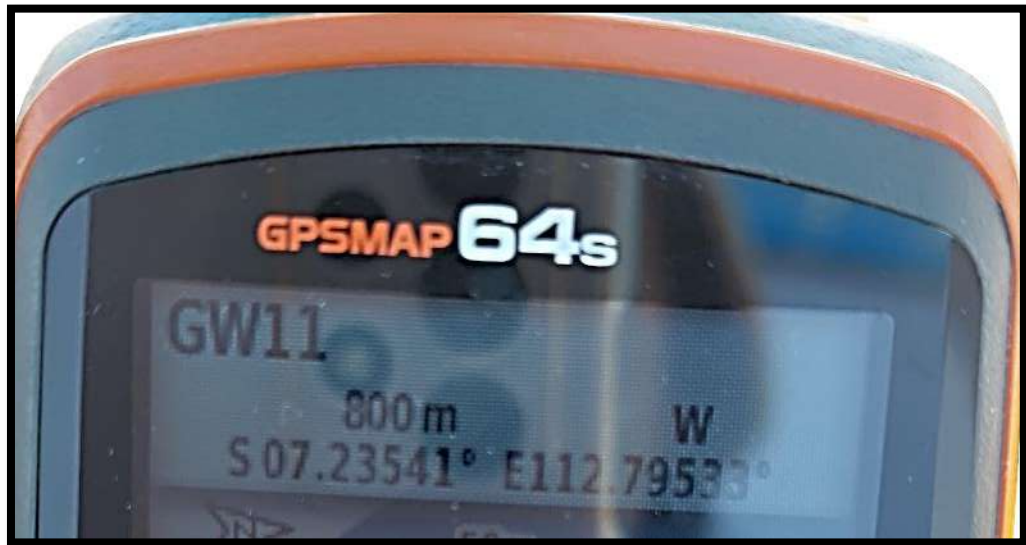
Tabel 4.18 menampilkan hasil pengujian dari perangkat *end node 2* menggunakan metode haversine pada sesi ketiga.

Tabel 4. 18 Hasil Pengujian *end node 2* sesi 3

Jarak (m)	RSSI	SNR	SF	Airtime (s)
602,07	-101	-5,25	10	0,33
602,01	-101	-5	10	0,33
601,76	-104	-7,5	10	0,33

Hasil dari pengujian pada *end node 2* sesi tiga menghasilkan rata-rata jarak sebesar 602,75 meter, dengan rata-rata RSSI sebesar -102,22, rata-rata SNR sebesar -7,11, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,29s sampai 0,33s.

Gambar 4.24 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada *end node 2* sesi empat.



Gambar 4. 24 Jarak *end node 2* Sesi 4 dengan GPS Garmin

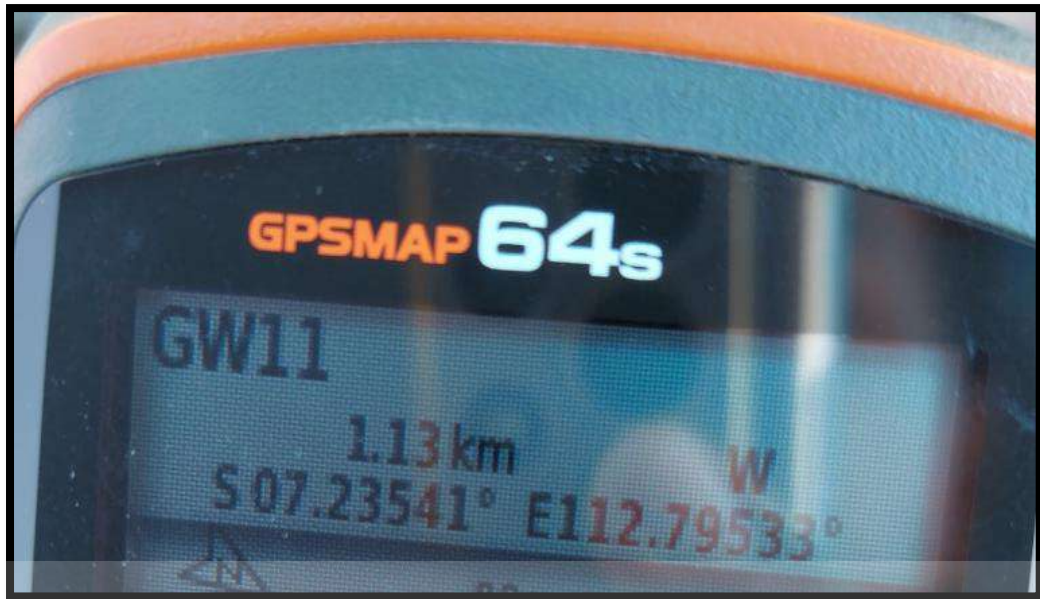
Tabel 4.19 menampilkan hasil pengujian dari perangkat *end node 2* menggunakan metode haversine pada sesi keempat.

Tabel 4. 19 Hasil Pengujian *end node 2* sesi 4

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
794,51	-107	-13,5	10	0,33
794,47	-102	-9,25	10	0,33
794,31	-106	-10,25	10	0,33

Hasil dari pengujian pada *end node 2* sesi empat menghasilkan rata-rata jarak sebesar 794,34 meter, dengan rata-rata RSSI sebesar -101,71, rata-rata SNR sebesar -8,39, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,33s.

Gambar 4.25 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada *end node 2* sesi lima.



Gambar 4. 25 Jarak *end node 2* Sesi 5 dengan GPS Garmin

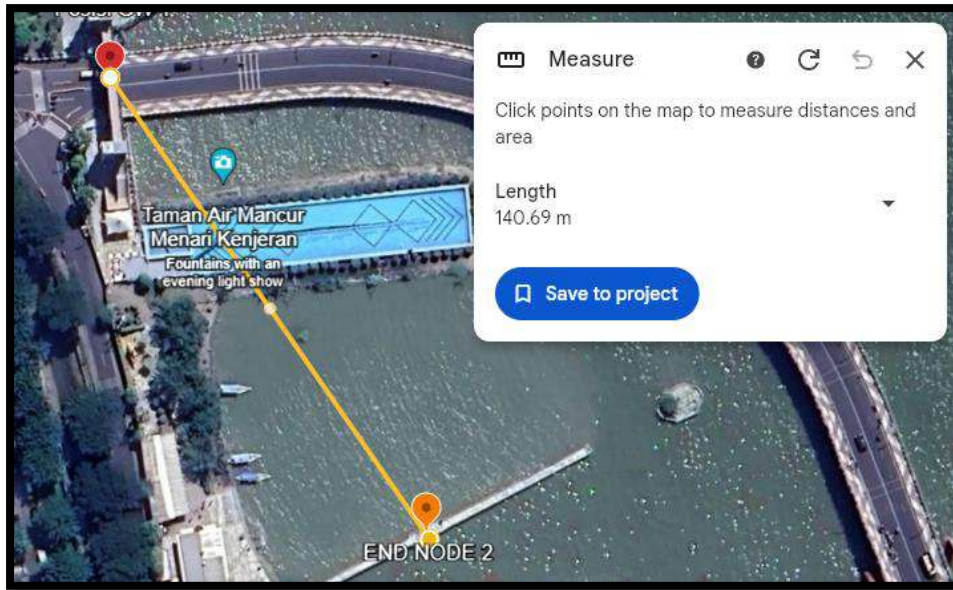
Tabel 4.20 menampilkan hasil pengujian dari perangkat *end node 2* menggunakan metode haversine pada sesi kelima.

Tabel 4. 20 Hasil Pengujian *end node 2* sesi 5

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
1136,93	-108	-14,5	10	0,33
1134,11	-110	-14,75	10	0,33
1132,55	-106	-15,25	10	0,33

Hasil dari pengujian pada *end node 2* sesi lima menghasilkan rata-rata jarak sebesar 1134,53 meter, dengan rata-rata RSSI sebesar -108, rata-rata SNR sebesar -14,83, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,33s.

3. Pengujian *end node 3*



Gambar 4. 26 Jarak *end node 3* dengan Google Earth

Tabel 4.21 menampilkan hasil dari pengujian *end node 3*, yang dimulai dari sesi 1 sampai 5.

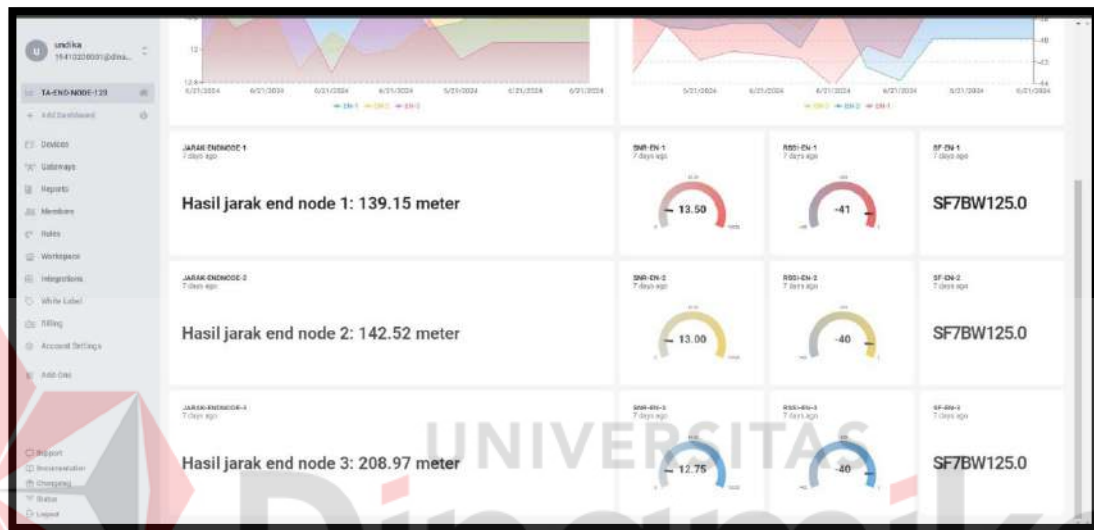
Tabel 4. 21 Hasil Pengujian *end node 3* Skenario 2

Sesi	Jarak (m)	RSSI	SNR	SF	Airtime (s)
1	141,35	-74	9,25	7	0,05
	141,35	-73	6,5	7	0,05
	141,35	-74	7,25	7	0,05
2	142,53	-73	4,75	10	0,37
	140,34	-73	4	10	0,37
	140,49	-77	7,75	7	0,05
3	141,30	-73	8,75	7	0,05
	142,00	-68	8,75	7	0,05
	142,10	-71	10,25	8	0,10
4	140,24	-74	6,5	8	0,10
	140,06	-77	7,75	9	0,18
	141,32	-70	7,25	9	0,16
5	143,26	-69	8,5	7	0,05
	143,26	-78	8,25	7	0,05
	143,26	-66	10,75	7	0,05

Hasil dari pengujian pada titik *end node* 3 menghasilkan rata-rata jarak sebesar 141,26 meter, dengan rata-rata RSSI sebesar -72,73, rata-rata SNR sebesar 7,56, dan nilai SF 7 sampai 10 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,05s hingga 0,37s.

4. Tampilan *Datacake*

Gambar 4.27 menampilkan dari hasil visualisasi pada *dashboard datacake*.



Gambar 4. 27 Hasil Visualisasi Pada *Dashboard Datacake*

4.2.6 Analisis Data

Tabel 4. 22 Hasil Analisis Pengujian Skenario Kedua

End Node	Sesi	Rata-rata RSSI (dBm)	Rata-rata SNR (dBm)	<i>Airtime</i> (s)	SF	<i>Haversine</i> (meter)	Selisih Jarak (meter)	Persentase Error %
1	1-5	-85,15	4,06	0,16	9	104,62	0,244	0,23%
	1	-88,56	0,22	0,33	10	241,28	3,276	1,38%
	2	-93,33	-1,46	0,33	10	429,28	2,280	0,53%
2	3	-102,22	-7,11	0,33	10	602,75	1,254	0,21%
	4	-101,71	-8,39	0,33	10	794,34	5,657	0,71%
	5	-108	-14,83	0,33	10	1134,53	4,530	0,40%
3	1-5	-72,73	7,56	0,11	7	141,26	0,570	0,41%

Tabel 4. 23 menunjukkan hasil pengujian yang dilakukan, bahwa RSSI dari ketiga *end node* mempunyai perubahan sinyal yang baik dengan rentang -72,73 hingga -102,22. Besar sinyal RSSI pada pengujian ini mengikuti dari jarak dari proses pengujian, hanya ada satu titik yang mengalami penurunan RSSI yaitu pada pengujian pada *end node* 2 sesi ke 4, secara keseluruhan data diatas membuktikan semakin jauh jarak pengujian maka semakin menurun kualitas sinyal yang diterima atau nilai RSSI yang diperoleh semakin melemah.

Nilai SNR yang dihasilkan mempunyai perubahan sinyal yang baik dengan rentang 0,22 hingga -14,83, semakin besar nilai SNR maka semakin baik komunikasi yang dilakukan, berdasarkan data diatas semakin jauh titik dari *end node* maka semakin kecil nilai SNR yang dihasilkan, sehingga menyebabkan banyak *noise* sinyal yang dihasilkan.

SF yang dihasilkan mempunyai perubahan nilai dari 7 hingga 10, semakin tinggi nilai SF maka akan memperlambat laju data tetapi memungkinkan komunikasi yang lebih andal dalam kondisi sinyal yang buruk atau pada jarak yang lebih jauh.

Nilai *Airtime* menunjukkan berapa lama data dapat terkirim melalui transmisi LoRa, berdasarkan data diatas menghasilkan rentang nilai *Airtime* 0,11s sampai 0,37s semakin jauh posisi dari *end node* maka semakin besar nilai *airtime* yang dihasilkan.

Hasil analisis dari pengujian komunikasi *multi node* menggunakan metode *haversine* pada skenario kedua dapat dikatakan cukup akurat dengan nilai presentasi *error* di bawah 1,4%.

4.3 Pengujian *End Node* Pada Kondisi Padat Pengunjung

4.3.1 Tujuan

Pengujian pengiriman data dilakukan saat kondisi padat pengunjung untuk mengukur ketahanan sinyal jika kondisi lingkungan tidak bisa selalu LOS (*Line of Sight*), dimana sistem yang dibuat dapat membuktikan komunikasi menggunakan transmisi LoRa dapat dilakukan pada kondisi lingkungan memiliki banyak halangan.

4.3.2 Peralatan yang Digunakan

1. 1 perangkat *end node* (Arduino, modul GPS, dan LoRa).
2. 1 modul *gateway* sebagai perantara antara *end node* dan server TTN.
3. 1 Laptop untuk pengambilan data pada *receiver*.

4.3.3 Cara Pengujian

1. *Mengupload* program pengiriman data jarak *haversine* dengan transmisi LoRa yang telah dibuat pada mikrokontroler Arduino.
2. Melakukan pengujian pada tiap titik dan jarak yang sudah ditentukan.
3. Menyimpan hasil pengiriman data yang ada pada server TTN.
4. Mengamati hasil dari pengujian yang telah dilakukan.

4.3.4 Skenario Pengujian Ketiga



Gambar 4. 28 Skenario Pengujian Ketiga

Gambar 4.28 menunjukkan skema posisi dari skenario pengujian ketiga, dimana lokasi *gateway* berpindah pada area yang ramai pengunjung. Pada pengujian kali ini perangkat yang digunakan untuk mengirimkan data hanya menggunakan 1 *end node* yang posisinya berada di laut. Terdapat 5 titik posisi *end*

node yang akan diuji, Tabel 4.32 menampilkan jarak pengujian dari alat GPS Garmin yang akan dilakukan.

Tabel 4. 24 Jarak GPS Garmin Pengujian Ketiga

Titik	Jarak (Garmin)
1	205 meter
2	400 meter
3	602 meter
4	801 meter
5	1010 meter

Gambar 4.29 menunjukkan posisi *gateway* pada pengujian ketiga yang akan dilakukan.



Gambar 4. 29 Posisi *Gateway* Skenario Pengujian Ketiga

Gambar 4.30 menunjukkan posisi perangkat *end node* pada pengujian ketiga yang berada di atas perahu.



Gambar 4. 30 Posisi Perangkat *End Node*

4.3.5 Hasil Pengujian

Pengujian dilakukan dengan pengambilan data yang diterima oleh *gateway* dari masing-masing posisi yang sudah ditentukan, dibawah ini adalah hasil dari data pengujian yang telah diperoleh dari posisi 1 sampai 5.

1. Posisi 1 (205m)

Gambar 4.31 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin, alat ini digunakan sebagai pembandingan dengan jarak yang dikirimkan oleh *end node* menggunakan metode *haversine*.



Gambar 4. 31 Jarak *end node* Sesi 1 dengan GPS Garmin

Tabel 4.24 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi pertama.

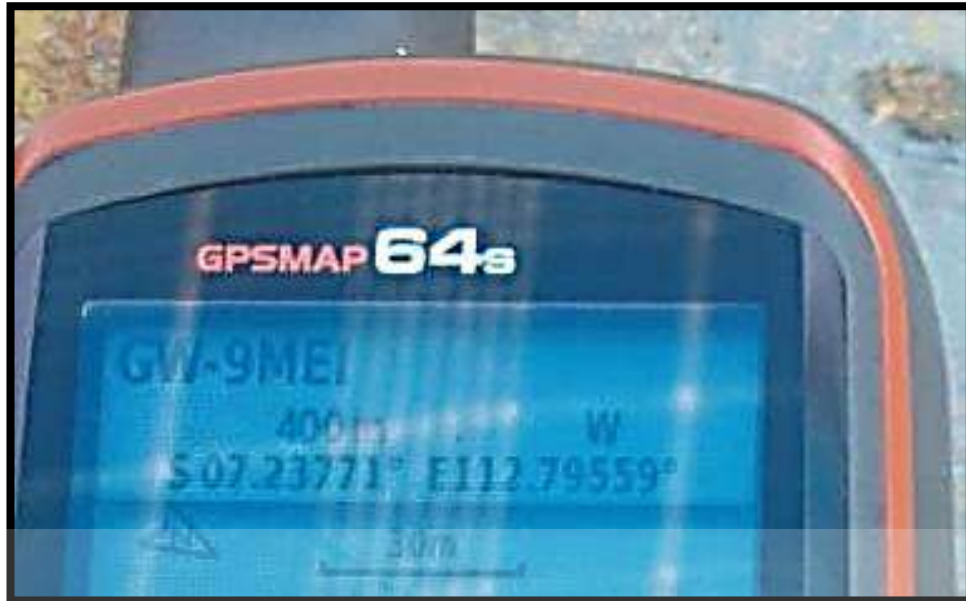
Tabel 4. 25 Hasil Pengujian *end node* posisi Pertama

Jarak (m)	RSSI	SNR	SF	Airtime (s)
205,52	-90	3	10	0,33
203,72	-86	3,75	10	0,33
205,37	-84	4,5	10	0,33
205,75	-89	2,5	10	0,33
209,00	-85	4	10	0,33

Hasil dari pengujian *end node* pada posisi pertama menggunakan metode *haversine* menghasilkan rata-rata jarak sebesar 213,42 meter, dengan rata-rata RSSI sebesar -88,89, rata-rata SNR sebesar 2,94, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,33s.

2. Posisi 2 (400m)

Gambar 4.32 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada posisi kedua.



Gambar 4. 32 Jarak *end node* Sesi 2 dengan GPS Garmin

Tabel 4.25 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi kedua.

Tabel 4. 26 Hasil Pengujian *end node* posisi kedua

Jarak (m)	RSSI	SNR	SF	Airtime (s)
401,12	-99	0,5	9	0,19
400,55	-93	3,25	9	0,19
429,48	-87	6,25	10	0,33
404,36	-98	1	10	0,33
405,24	-95	2	10	0,33

Hasil dari pengujian *end node* pada posisi kedua menggunakan metode *haversine* menghasilkan rata-rata jarak sebesar 404,23 meter, dengan rata-rata RSSI sebesar -93,90, rata-rata SNR sebesar 1,98, rata-rata SNR sebesar 2,94, dan nilai SF 9 sampai 10 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,19 sampai 0,33s.

3. Posisi 3 (602m)

Gambar 4.33 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada posisi ketiga.



Gambar 4. 33 Jarak *end node* Sesi 3 dengan GPS Garmin

Tabel 4.26 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi ketiga.

Tabel 4. 27 Hasil Pengujian *end node* posisi Ketiga

Jarak (m)	RSSI	SNR	SF	Airtime (s)
608,01	-91	4,25	9	0,19
608,01	-91	5,5	9	0,19
605,15	-91	2,5	9	0,19
605,20	-92	3,25	9	0,19

Hasil dari pengujian *end node* pada posisi ketiga menggunakan metode *haversine* menghasilkan rata-rata jarak sebesar 604,12 meter, dengan rata-rata RSSI sebesar -93,29, dan rata-rata SNR sebesar 3,48, dan nilai SF 9 sampai 10 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,19 sampai 0,33s.

4. Posisi 4 (801m)

Gambar 4.34 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada posisi keempat.



Gambar 4. 34 Jarak *end node* Sesi 4 dengan GPS Garmin

Tabel 4.27 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi keempat.

Tabel 4. 28 Hasil Pengujian *end node* posisi Keempat

Jarak (m)	RSSI	SNR	SF	<i>Airtime</i> (s)
804,66	-100	-0,25	10	0,33
803,90	-107	-1,25	10	0,33
801,47	-101	-1,25	10	0,33
803,49	-102	1,25	10	0,33
803,49	-100	0,25	10	0,33

Hasil dari pengujian *end node* pada posisi keempat menggunakan metode *haversine* menghasilkan rata-rata jarak sebesar 803,22 meter, dengan rata-rata RSSI sebesar -104,22, dan rata-rata SNR sebesar -1,94, dan nilai SF 9 sampai 10 sehingga menghasilkan waktu dari parameter *Airtime* 0,33s.

5. Posisi 5 (1010m)

Gambar 4.35 menunjukkan hasil pengujian jarak menggunakan alat GPS garmin pada posisi kelima.



Gambar 4. 35 Jarak *end node* Sesi 5 dengan GPS Garmin

Tabel 4.28 menampilkan hasil pengujian dari perangkat *end node* menggunakan metode *haversine* pada posisi kelima.

Tabel 4. 29 Hasil Pengujian *end node* posisi Kelima

Jarak (m)	RSSI	SNR	SF	Airtime (s)
1000,97	-101	-1	10	0,33
1002,72	-99	0,5	10	0,37
1002,95	-99	-0,5	10	0,37
1005,26	-99	-2	10	0,37
1006,88	-102	0,25	10	0,37

Hasil dari pengujian *end node* pada posisi kelima menggunakan metode *haversine* menghasilkan rata-rata jarak sebesar 1004,80 meter, dengan rata-rata RSSI sebesar -100,14, dan rata-rata SNR sebesar -1,00, dan nilai SF 10 sehingga menghasilkan waktu dari parameter *Airtime* dengan rentang 0,33s sampai 0,37s.

4.3.6 Analisis Data

Tabel 4.29 menampilkan hasil analisis pengujian dari perangkat *end node* menggunakan metode *haversine*, dimulai dari posisi 1 sampai 5 yang telah dilakukan.

Tabel 4. 30 Hasil Analisis Pengujian Skenario Ketiga

Titik	Rata-Rata RSSI	Rata-Rata SNR	<i>Airtime</i> (s)	SF	<i>Haversine</i> (meter)	Selisih Jarak (meter)	Persentase Error %
1	-88,89	2,94	0,33	10	213,42	8,43	4,11%
2	-94,67	1,98	0,31	10	404,23	4,24	1,06%
3	-93,29	3,48	0,19	9	604,11	2,12	0,35%
4	-104,22	-1,94	0,28	10	803,16	2,11	0,27%
5	-100,14	-1,00	0,37	10	1004,80	5,20	0,51%

Dilihat dari hasil pengujian yang dilakukan, RSSI menghasilkan perubahan sinyal yang baik dengan rentang -88,89 hingga -100,14. Besar sinyal RSSI pada pengujian ini mengikuti dari jarak dari proses pengujian. Terdapat penurunan sinyal RSSI pada titik 3 dan 5, tidak konsistennya sinyal RSSI juga dipengaruhi dari kondisi area pengiriman data, dimana pada pengujian ini dilakukan pada kondisi yang ramai pengunjung dan juga terdapat beberapa halangan seperti pohon, bangunan, dan jembatan.

Nilai SNR yang dihasilkan mempunyai perubahan sinyal yang baik dengan rentang 2,94 hingga -1,94, semakin besar nilai SNR maka semakin baik komunikasi yang dilakukan, berdasarkan data diatas semakin jauh titik dari *end node* maka semakin kecil nilai SNR yang dihasilkan, sehingga menyebabkan banyak *noise* sinyal yang dihasilkan.

SF yang dihasilkan mempunyai perubahan nilai dari 9 hingga 10, semakin tinggi nilai SF maka akan memperlambat laju data tetapi memungkinkan komunikasi yang lebih andal dalam kondisi sinyal yang buruk atau pada jarak yang lebih jauh.

Nilai *Airtime* menunjukkan berapa lama data dapat terkirim melalui transmisi LoRa, berdasarkan data diatas menghasilkan rentang nilai *Airtime* 0,19s sampai

0,37s semakin jauh posisi dari *end node* maka semakin besar nilai *airtime* yang dihasilkan.

Sedangkan hasil analisis dari pengujian komunikasi *multi node* menggunakan metode *haversine* pada skenario kedua dapat dikatakan cukup akurat dengan nilai presentasi *error* di bawah 4,5%.



UNIVERSITAS
Dinamika

BAB V

KESIMPULAN

5.1 Kesimpulan

Dari beberapa sesi pengujian yang telah dilakukan, terdapat beberapa kesimpulan yang didapat, yaitu:

1. Data yang dikirimkan oleh *end node* berhasil diterima oleh *gateway* dengan beberapa skenario pengujian. Pengujian skenario pertama, dimana *end node* mengirimkan data jarak menggunakan metode *haversine* pada 10 titik posisi yang berbeda, menghasilkan rentang *error* di bawah 5%. Pengujian skenario kedua, jumlah *end node* yang digunakan adalah 3. Pengiriman data dari *end node* dilakukan secara bersamaan, menghasilkan rentang *error* dibawah 1,4%. Dan pengujian skenario ketiga, Pengujian komunikasi tidak hanya dilakukan pada kondisi tanpa halangan, namun juga pada kondisi yang ramai pengunjung dan terdapat halangan seperti pohon dan bangunan lain. Pengujian dengan kondisi yang ramai menghasilkan rentang *error* di bawah 4,5%.
2. Analisa performa komunikasi pada pengujian area pesisir pantai menghasilkan beberapa parameter hasil pengujian yaitu RSSI, SNR, *Airtime*, dan *Spreading Factor*. Pada pengujian pertama sinyal RSSI yang dihasilkan mempunyai perubahan sinyal dengan rentang -51,33 hingga -105,67, pengujian kedua dengan rentang -72,73 hingga -102,22, dan pengujian ketiga dengan rentang -88,89 hingga -100,14. Sinyal RSSI yang diperoleh dapat mengikuti perubahan dari jarak yang diuji, semakin jauh jarak pengujian, maka semakin melemah nilai sinyal RSSI yang diperoleh. SNR pada pengujian pertama menghasilkan perubahan sinyal *noise* rentang 13,42 hingga -10,17, pengujian kedua dengan rentang 0,22 hingga -14,83, dan pengujian ketiga dengan rentang 2,94 hingga -1,94. Nilai SNR yang dihasilkan menunjukkan perubahan sesuai dengan jarak pengujian, semakin jauh jarak pengujian maka *noise* yang diperoleh akan semakin besar. *Spreading Factor* pada pengujian pertama dan kedua menghasilkan perubahan nilai dari 7 hingga 10, sedangkan pengujian ketiga menghasilkan perubahan nilai dari 9 hingga 10. Semakin tinggi nilai SF maka

akan mempengaruhi waktu *Airtime*, tetapi memungkinkan komunikasi yang lebih andal dalam kondisi sinyal yang buruk atau pada jarak yang lebih jauh. Dan *Airtime* pada pengujian Pertama menghasilkan rentang nilai 0,07s sampai 0,37s, pengujian kedua menghasilkan rentang nilai *Airtime* 0,11s sampai 0,37s, dan pengujian ketiga menghasilkan rentang nilai *Airtime* 0,19s sampai 0,37s, nilai *Airtime* yang diperoleh pada pengujian yang dilakukan menunjukkan bahwa semakin jauh jarak perangkat *end node* maka semakin besar nilai *Airtime* yang dihasilkan.

5.2 Saran

Saran dibuat untuk agar penelitian selanjutnya mengenai komunikasi *multi node* menggunakan komunikasi LoRa pada area pesisir pantai bisa lebih baik lagi. Saran untuk penelitian yang ingin dilakukan selanjutnya adalah:

1. Menambahkan jumlah *end node* agar perangkat *gateway* bisa lebih dimaksimalkan, dalam pengimplementasian alat secara langsung.
2. Menambahkan jumlah *gateway* agar bisa menerima perangkat *end node* yang lebih banyak dan bisa menerapkan sistem *multi gateway*.

DAFTAR PUSTAKA

- Aduino.S.r.l. (2024, January 2). *store.arduino.cc/products/arduino-due*. Retrieved from store.arduino.cc: <https://store.arduino.cc/products/arduino-due>
- Affandi, C. D. (2020). *Rancang Bangun Sistem Keamanan Gudang Penyimpanan Menggunakan Transmisi LoRa*. Surabaya: Universitas Dinamika.
- AG, u.-b. (2024, January 2). *NEO-6 Datasheet by SparkFun Electronics*. Retrieved from [digikey.com/htmldatasheets/production/2757457/0/0/1/neo-6.html](https://www.digikey.com/htmldatasheets/production/2757457/0/0/1/neo-6.html): <https://www.u-blox.com/en/>
- Alliance, L. (2024, January 2). *THE THINGS NETWORK*. Retrieved from [thethingsnetwork.org](https://www.thethingsnetwork.org/): <https://www.thethingsnetwork.org/country/indonesia/>
- Alliance, L. (2024, January 7). *what is LoRaWAN spesification*. Retrieved from [lora-alliance.org: https://lora-alliance.org/about-lorawan/](https://lora-alliance.org/about-lorawan/)
- Blenn, N., & Kuipers, F. (2017). LoRaWAN in the Wild. *Measurements from The Things Network*, 1-9.
- Budi Setiadi, R. S. (2023). Estimasi Jarak pada Sistem Koordinat Berbasis Metode Haversine Menggunakan Tapis Kalman. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 207 - 216.
- Fahmida, S., Modekurthy, V. P., Rahman, M., & Saifullah, A. (2023). Handling Coexistence of LoRa with Other Networks through Embedded Reinforcement Learning. *IoTDI*, 1-14.
- Harjowinoto, F. S. (2021). *Analisis Transmisi Data LoRa pada Mobile Node*. Surabaya: Universitas Dinamika.
- Hasanah, M., & Suharso, A. (2023). Algoritma Haversine pada Sistem Informasi Geografis: Tinjauan Literatur Sistematis. *NUANSA INFORMATIKA*, 1-9.
- KOMINFO, D. (2024, January 1). *WILAYAH PESISIR PANTAI MASIH KURANG PENDIRIAN AKSES INTERNET*. Retrieved from kominfo.jatimprov.go.id: <https://kominfo.jatimprov.go.id/read/umum/17650>
- Leibbrandt, A. (2024, January 2). *An Introduction to LoRa Gateways*. Retrieved from blog.akenza.io/lora-gateway-intro: <https://blog.akenza.io/lora-gateway-intro>

Nugraha, A. P., Ikasari, D., & Widiastuti. (2023). PENCARIAN TOKO OBAT HERBAL HNI TERDEKAT MENGGUNAKAN FORMULA HAVERSINE. *Engineering and Science*, 1-8.

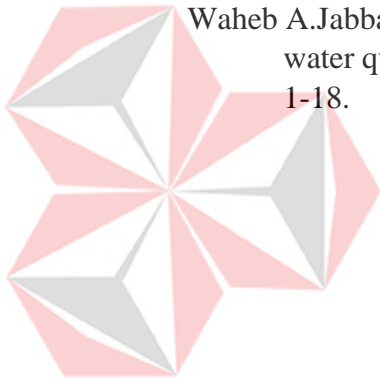
OKKIE PUSPITORINI, H. M. (2023). implementasi telemetry dan evluasi perfomansi sistem komunikasi lora di daerah pesisir pantai. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 180 - 194.

Sari, D. I., Suendri, & Putri, R. A. (2023). Penerapan Metode Haversine Pada Lokasi Rawan Kriminalitas dan Kecelakaan di Kota Medan Berbasis Android. *Jurnal Ilmu Komputer*, 1-11.

setiawan, I. p. (2020). *Analisis Parameter LoRa Pada Lingkungan Indoor*. Surabaya: Universitas Dinamika.

Thethingsnetwork. (2024, january 7). *The LoRaWAN Architecture*. Retrieved from thethingsnetwork:
<https://www.thethingsnetwork.org/docs/lorawan/architecture/>

Waheb A.Jabbar, T. M. (2024). Development of LoRaWAN-based IoT system for water quality monitoring in rural areas. *Expert Systems With Applications*, 1-18.



UNIVERSITAS
Dinamika