

PENGEMBANGAN EWS BANJIR DENGAN TEKNOLOGI LORA

Tugas Akhir

Disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Ahli Madya
pada Program Diploma III Teknik Informatika

Universitas Sebelas Maret



Disusun Oleh :

WAHYU KRISTIYANTO

M3117093

PROGRAM DIPLOMA III TEKNIK INFORMATIKA

SEKOLAH VOKASI

UNIVERSITAS SEBELAS MARET

SURAKARTA

2020

PERSETUJUAN

PENGEMBANGAN EWS BANJIR DENGAN TEKNOLOGI LORA

Disusun Oleh :

WAHYU KRISTIYANTO

M3117093

Disetujui untuk dipertahankan dihadapan Panitia Ujian Tugas Akhir

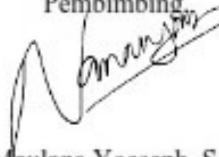
Program Studi Diploma III Teknik Informatika

Sekolah Vokasi

Universitas Sebelas Maret Surakarta

Surakarta, 10 Juli 2020

Pembimbing



Nanang Maulana Yoeseph, S.Si., M.Cs.
NIP. 1981071420160601

PENGESAHAN
PENGEMBANGAN EWS BANJIR DENGAN TEKNOLOGI LORA

Tugas Akhir ini telah diuji dan disahkan oleh Panitia Ujian Tugas Akhir
Program Studi Diploma III Teknik Informatika
Sekolah Vokasi
Universitas Sebelas Maret Surakarta

Pada Hari : Rabu
Tanggal : 22 Juli 2020

Disusun Oleh :
WAHYU KRISTIYANTO
M3117093

Panitia Ujian Tugas Akhir Nama

Tanda Tangan

1. Penguji 1

Nanang Maulana Y. S.Si., M.Cs.
NIK. 1981071420160601

2. Penguji 2

Rudi Hartono, S.Si., M.Eng
NIK. 1980082920130201

3. Penguji 3

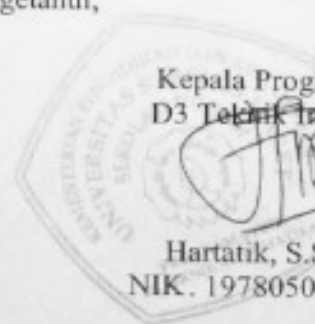
Muhammad Asri Safi'ie S.Si., M.Kom
NIK. 1981110320180601

Mengetahui,



Direktur
Sekolah Vokasi

Dr. H. Tri Hananto, M.Acc., Ak.
NIK. 196909241994021001



Kepala Program Studi
D3 Teknik Informatika

Hartatik, S.Si., M.Si.
NIK. 1978050320130201

HALAMAN PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Laporan Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar akademik di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam Daftar Pustaka.

Surakarta, 1 Maret 2020

Wahyu Kristiyanto

M3117093

HALAMAN MOTTO

MOTTO :

- Tanpa cinta, kecerdasan itu berbahaya. Ilmu pengetahuan, iman dan taqwa harus bersatu. (Bacharuddin Jusuf Habibie)

HALAMAN PERSEMBAHAN

Tugas akhir ini dipersembahkan untuk :

1. Keluarga tercinta
2. Dosen pembimbing Pak Nanang Maulana Yoesep
3. Teman teman D3 Teknik Informatika.

KATA PENGANTAR

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa atas limpahan rahmat-Nya, sehingga dapat menyelesaikan Laporan Tugas Akhir dengan judul “Pengembangan EWS Banjir Dengan Teknologi LoRa”. Penyusunan laporan ini bertujuan untuk memenuhi persyaratan gelar ahli madya Program Diploma III Teknik Informatika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta. Dalam penyusunan Laporan Tugas Akhir ini penulis mendapatkan bantuan dan dukungan dari berbagai pihak. Semoga Tuhan Yang Maha Esa memberikan balasan yang lebih baik kepada pihak-pihak yang telah memberikan bantuan dan dukungan dari awal kegiatan magang sampai penyusunan Laporan Kegiatan Magang Mahasiswa ini selesai. Semoga dengan selesainya penyusunan Laporan Kegiatan Magang Mahasiswa ini dapat memberikan manfaat dan kebaikan pada penulis dan pembaca sekalian.

Surakarta, 1 Maret 2020

Wahyu Kristiyanto

DAFTAR ISI

PERSETUJUAN	ii
PENGESAHAN	iii
HALAMAN PERNYATAAN	iv
HALAMAN MOTTO	v
HALAMAN PERSEMBAHAN	vi
KATA PENGANTAR	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR	xi
INTISARI.....	xiii
ABSTRACT	xiv
BAB I	15
PENDAHULUAN	15
1.1. Latar Belakang	15
1.2. Perumusan Masalah.....	16
1.3. Tujuan.....	16
1.4. Manfaat.....	17
1.5. Batasan Masalah	17
1.6. Metodologi	17
1.7. Sistematika Penulisan.....	17
BAB II.....	19
TINJAUAN PUSTAKA DAN LANDASAN TEORI	19
2.1. Tinjauan Pustaka	19
2.2. Landasan Teori	20
2.3. Sistem Peringatan Dini Banjir	24

2.4. Teknologi LoRa.....	25
BAB III	26
ANALISIS DAN RANCANGAN SISTEM	26
3.1. Alat Dan Bahan	26
3.2. Alur Kerja Sistem.....	27
3.3. Desain Skematik.....	28
BAB IV	29
IMPLEMENTASI DAN PENGUJIAN	29
4.1. Implementasi	29
4.2. Pengujian	47
BAB V.....	53
KESIMPULAN DAN SARAN.....	53
5.1. Kesimpulan.....	53
5.2. Saran	53
DAFTAR PUSTAKA	54

DAFTAR TABEL

Tabel 4. 1 Data hasil pengujian.....	49
Tabel 4. 2 Perbedaan waktu antar layanan.....	50
Tabel 4. 3 Perbandingan Data Sensor Arus Air	50
Tabel 4. 4 Perbandingan Data Sensor Ultrasonic	51

DAFTAR GAMBAR

Gambar 2. 1 Sensor Ultrasonic SRF05	23
Gambar 2. 2 Sensor Waterflow YF-S201	24
Gambar 3. 1 Alur Kerja Sistem.....	27
Gambar 3. 2 Desain Skematik.....	28
Gambar 4. 1 Halaman Login Dragino.....	30
Gambar 4. 2 Konfigurasi LoRaWan Server.....	31
Gambar 4. 3 Konfigurasi Channel 1 Radio.....	32
Gambar 4. 4 Tampilan Jaringan Wireless	33
Gambar 4. 5 Halaman Join Network Dragino.....	34
Gambar 4. 6 Konfigurasi Jaringan Wireless	34
Gambar 4. 7 Wireless Overview	35
Gambar 4. 8 Halaman Registrasi Gateway	36
Gambar 4. 9 Halaman Gateway Overview	37
Gambar 4. 10 Halaman Add Application.....	38
Gambar 4. 11 Application Overview	39
Gambar 4. 12 Halaman Registrasi Device	39
Gambar 4. 13 Tampilan Device EUI, App EUI dan App Key.....	40
Gambar 4. 14 Halamana Penambahan HTTP Integration.....	41
Gambar 4. 15 Integration Overview.....	42
Gambar 4. 16 Tampilan Konfigurasi File Config	43
Gambar 4. 17 Sketch : pendefinisian variabel	44
Gambar 4. 18 Sketch : Device EUI, Application EUI dan App Key	45
Gambar 4. 19 Sketch : fungsi loop sensor waterflow	45

Gambar 4. 20 Sketch : Fungsi pengiriman data	46
Gambar 4. 21 Rangkaian device untuk pengujian I	47
Gambar 4. 22 Rangkaian device untuk pengujian II.....	48

INTISARI

Penelitian ini ditujukan untuk mengembangkan sistem peringatan dini untuk banjir dengan menggunakan teknologi LoRa. Fenomena bencana banjir yang sering terjadi pada area sekitar sungai ataupun perkotaan membuat kebutuhan untuk sebuah sistem peringatan dini. Penelitian ini dilakukan dengan menggunakan sensor pengukuran jarak dan arus air yang didesain dalam jaringan teknologi LoRa. Penggunaan teknologi LoRa dipilih karena kelebihanannya yang berdaya rendah tetapi dengan jangkauan yang luas. Dengan menggunakan teknologi ini, uji coba dapat dilaksanakan dengan lebih efektif dan efisien. Hasil dari penelitian ini menunjukkan sistem peringatan dini yang dibangun berfungsi beserta dengan pengiriman data yang dilakukan dengan menggunakan teknologi LoRa. Data dari hasil pengembangan ini kemudian dapat digunakan untuk pembangunan aplikasi monitoring kondisi sungai yang rawan akan terjadi bencana banjir.

Kata Kunci : Sistem Peringatan Dini, Banjir, LoRa.

ABSTRACT

This study aims to develop an early warning system for floods using LoRa technology. The phenomenon of flood disasters that often occurs in areas around rivers or cities makes the need for an early warning system. This research was conducted using a sensor for measuring distance and water flow which is designed in the LoRa technology network. The use of LoRa technology was chosen because of its low power but wide coverage. By using this technology, trials can be carried out more effectively and efficiently. The results of this study indicate that the early warning system built functions along with data transmission using LoRa technology. The data from the results of this development can then be used for the construction of a river monitoring application that is prone to flooding.

Keywords: Early Warning System, Flood, LoRa.

BAB I

PENDAHULUAN

1.1. Latar Belakang

Sebagai sebuah negara yang dikaruniai dengan kekayaan lingkungan yang besar, tersimpan sebuah bahaya yang selalu siap untuk menghantam Indonesia. Kekayaan lingkungan termasuk didalamnya kekayaan hayati yang disatu sisi membawa sumber daya alam yang melimpah, disisi lain membawa dampak yang bersifat menghancurkan yakni bencana alam(*natural disaster*). Bencana alam merupakan sesuatu yang tidak langka lagi dijumpai di wilayah Indonesia. Mulai dari bencana alam seperti banjir, ledakan gunung berapi, gempa bumi, tsunami, dan lain sebagainya. Bencana-bencana tersebut kerap menimpa mengingat kondisi geografis Indonesia yang memang sangat memungkinkan untuk terjadinya hal tersebut. Indonesia mencatatkan banyak bencana alam, pada tahun 2019 saja tercatat terdapat 9391 bencana alam menurut data dari BNPB.

Meskipun begitu, dalam hal penanganan bencana alam Indonesia masih dinilai kurang. Contoh paling nyata dari hal ini ialah banjir yang kerap menyambangi kota-kota besar. Begitu lama banjir menjadi masalah yang menghantui wilayah perkotaan besar. Menurut data yang dihimpun dari Badan Nasional Penganggulangan Bencana atau BNPB, dalam kurun waktu 2001 sampai dengan 2019 terdapat rata-rata 775 bencana banjir setiap tahunnya.

Upaya pencegahan dan penanganan yang baik terkait bencana banjir diperlukan guna meminimalkan resiko korban jiwa dan kerugian yang dapat dialami.

Pencegahan banjir diantaranya dapat diberikan arahan kepada masyarakat luas untuk menjaga lingkungan dengan tidak membuang sampah sembarangan. Hal ini bergantung pada masyarakat itu sendiri apakah patuh atau tidak terhadap aturan yang dibuat. Hal lain yang dapat dilakukan untuk dapat meminimalkan resiko jika akan terjadi banjir ialah dengan mengembangkan sebuah sistem peringatan dini untuk mengingatkan pemerintah dalam menangani banjir dan masyarakat agar bersiap untuk menghadapi banjir.

Pengembangan sistem ini membutuhkan teknologi yang dapat diandalkan guna mencapai hasil yang maksimal. Teknologi LoRa merupakan protokol jaringan berdaya rendah dengan cakupan area yang cukup luas. Dalam pengembangan sebuah sistem yang terkait dengan *Internet of Things*, teknologi LoRa cocok untuk digunakan. Hal ini karena teknologi LoRa memiliki kelebihan dibandingkan teknologi lain yang ada. Jangkauan jarak yang jauh dengan konsumsi daya yang rendah adalah kelebihan utama yang dimiliki LoRa. Pengembangan dengan menggunakan teknologi LoRa dibutuhkan untuk membangun sistem peringatan dini banjir yang lebih baik.

1.2. Perumusan Masalah

Untuk dapat meminimalkan resiko dan kerugian yang dapat dialami dari bencana banjir, maka perlu dikembangkan sebuah perangkat teknologi sebagai sebuah sistem peringatan dini (*early warning system*). Maka dari itu rumusan masalah yang dapat diambil dalam laporan tugas akhir ini diantaranya :

- a. Bagaimana pengembangan sistem peringatan dini banjir menggunakan teknologi LoRa?

1.3. Tujuan

Dalam pembuatan tugas akhir ini memuat beberapa tujuan diantaranya :

- a. Mengembangkan sistem peringatan dini menggunakan teknologi LoRa.

1.4. Manfaat

Manfaat yang dapat diambil dari pembuatan tugas akhir ini diantaranya :

- a. Untuk dapat mengetahui pengembangan sistem peringatan dini menggunakan teknologi LoRa.
- b. Untuk dapat membantu mengatasi permasalahan banjir dengan menggunakan sistem peringatan dini banjir dengan teknologi LoRa.

1.5. Batasan Masalah

Pengembangan sistem peringatan dini banjir dengan teknologi LoRa ini mempunyai beberapa batasan masalah diantaranya :

- a. Sistem peringatan dini banjir ini mengambil 2 data dengan 2 buah sensor yakni sensor ultrasonic untuk ketinggian muka air dan sensor *water flow*.
- b. Jarak jangkauan untuk kedalaman air terbatas hanya sampai dengan 4,5 meter sesuai dengan spesifikasi dari sensor ultrasonic.

1.6. Metodologi

Metode yang digunakan dalam penelitian ini menggunakan metode kuantitatif dengan menggunakan data-data yang diperoleh dari pengujian. Hasil dari pengujian yang dilakukan kemudian diolah menjadi informasi untuk menampilkan keefektifan pengembangan sistem yang dilakukan.

1.7. Sistematika Penulisan

Penulisan laporan tugas akhir “**PENGEMBANGAN *EWS (EARLY WARNING SYSTEM)* BANJIR DENGAN TEKNOLOGI LORA**“ ini disusun secara sistematis dan dibagi menjadi beberapa bab.

Penulisan setiap bab pada laporan tugas akhir ini ditulis secara runtut dari BAB I sampai dengan BAB V dengan rincian isi sebagai berikut:

BAB I PENDAHULUAN

Pada bab pertama ini dijelaskan mengenai latar belakang, perumusan masalah, tujuan, manfaat, batasan masalah, metodologi serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI

Pada bab kedua ini dijelaskan mengenai tinjauan pustaka yang digunakan dan landasan teori yang mendukung penulisan laporan.

BAB III ANALISIS DAN RANCANGAN SISTEM

Pada bab ketiga mengenai analisis dan rancangan sistem dijelaskan mengenai alat dan bahan yang digunakan, alur kerja sistem dan perancangan desain hardware.

BAB IV IMPLEMENTASI DAN PENGUJIAN

Pada bab keempat dijelaskan mengenai implementasi hardware dan informasi tentang pengujian yang sudah dilakukan.

BAB V KESIMPULAN DAN SARAN

Pada bab kelima dijelaskan mengenai kesimpulan dari pengembangan yang sudah dilakukan serta saran untuk pengembangan selanjutnya.

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1. Tinjauan Pustaka

Terdapat penelitian sebelumnya yang sudah dilakukan dan memiliki kemiripan dengan penelitian ini.

Penelitian yang dilakukan oleh saudara Alif Putra Lestari yang membahas mengenai peringatan dini banjir bengawan solo di kabupaten tuban, jawa timur. Penelitian tersebut menggunakan arduino uno dengan sensor ultrasonic. Pada penelitian tersebut, kondisi ketinggian muka air dibagi menjadi 3 status siaga 1, 2 dan waspada. Sistem tersebut bekerja dengan menggunakan *buzzer* dan lampu *LED* untuk sebagai bentuk peringatan. Perangkat dipasang pada tiang yang terpasang secara permanen didekat sungai. Sistem tersebut memiliki keterbatasan pada jangkauan peringatan yang dimiliki. Masyarakat disekitar hanya dapat mengetahui peringatan bencana apabila ia berada pada jangkauan perangkat yang terpasang.

Dengan demikian maka dibutuhkan sistem peringatan yang dapat menerima data *realtime* sungai untuk dapat memantau perkembangan kondisi sungai. Data yang diterima diolah menjadi sebuah informasi yang dapat digunakan sebagai sistem peringatan dini pencegahan banjir. Dengan begitu penanganan atau antisipasi bencana dapat dilakukan dengan lebih baik dan cepat.

2.2. Landasan Teori

Dalam pembuatan tugas akhir ini dipaparkan teori yang menjadi landasan dalam penyusunan laporan dan pengembangan projek ini, diantaranya :

a. Sensor

Sensor adalah perangkat yang mendeteksi dan merespon masukan dari lingkungan fisik.(Adin, Bhawiyuga and Yahya, 2019) Sensor memiliki banyak varian dan tipe, dengan berbagai fungsi yang beraneka ragam. Masukan atau *input* yang dapat diterima sensor pun beragam, mulai dari cahaya, gerakan, panas, kelembaban, udara, air dan lain sebagainya.

b. LoRa

Menurut Adin, Bhawiyuga dan Yahya (2019), LoRa merupakan salah satu protokol *Low Power Wide Area Wireless Network (LPWAN)* untuk aplikasi *Internet of Things (IoT)*. LoRa mempunyai beberapa kelebihan salah satunya bisa mengakomodasi jarak mencapai 1 km lebih dengan pengaturan yang tepat dan juga lingkungan yang mendukung seperti sedikitnya gangguan serta jenis perangkat yang mendukung implementasi perangkat LoRa. (Susanto, Bhawiyuga and Amron, 2019) Disamping itu kelebihan yang lain ialah sesuai namanya yakni konsumsi daya yang rendah.

c. Gateway

Gateway merupakan perangkat yang dapat menghubungkan jaringan nirkabel LoRa dengan *IP Network* melalui *WiFi*, *Ethernet* maupun jaringan 3G/4G. Perangkat gateway LoRa memiliki metode koneksi internet yang bervariasi seperti *interface WiFi*, *port ethernet* dan *usb host port*. Semua metode tersebut memberikan keleluasaan bagi pengguna untuk dapat tersambung ke-internet sesuai dengan kebutuhan.

Gateway yang digunakan pada penelitian ini adalah gateway Dragino dengan model LG02. Gateway ini memiliki 400 Mhz ar331 *processor* dengan 64MB RAM dan 16MB Flash. Untuk *interface* gateway ini memiliki 2 port RJ45, Power Input 12 V DC, USB 2 *Host Connector*, USB 2.0 *Host internal interface*, 2 LoRa *Interface*, WiFi serta LoRa *Wireless*. Untuk LoRa, gateway ini memiliki 2 band frekuensi yakni HF(862 ~ 1020 Mhz) dan LF(410 ~ 528 Mhz) dengan modulasi FSK, GFSK, MSK, GMSK, LoRaTM dan OOK. Spesifikasi WiFi pada gateway ini yakni IEEE 802.11 b/g/n dengan frekuensi 2.4 ~ 2.46 GHz.

d. The Things Network

The Things Network merupakan platform penyedia layanan *IoT* (*Internet of Things*) yang didirikan di Belanda. Layanan ini bersifat global, *decentralized*, *open* dan *free* untuk jaringan *Internet of Things*. Platform ini memberikan fasilitas kepada pengguna untuk dapat mengelola jaringan perangkat serta gateway yang mereka miliki. Dengan menggunakan layanan The Things Network, pengguna dapat membangun jaringan iot terpadu yang dapat digunakan untuk berbagai hal.

e. Google Firebase

Merupakan layanan database yang dimiliki oleh Google. Layanan ini memberikan kemudahan untuk dapat membuat aplikasi secara realtime dengan lebih mudah. Google Firebase dapat digunakan sebagai aplikasi pihak ke tiga yang membantu mengelola aplikasi atau sistem yang dimiliki pengguna. Sebagai layanan yang bersifat praktis, Google Firebase memiliki berbagai fitur menarik yang dapat dengan mudah digunakan serta dengan tutorial yang sudah tersedia secara lengkap.

f. Lora Node

End Node yang digunakan untuk implementasi teknologi LoRa ini yakni BSFrance LoRa32u4 II v1.4. *Board* Lora32u4 ini merupakan kombinasi antara Atmel ATmega32u4 AVR MCU dan Lora Module serta memiliki dukungan baterai Li-Ion/LiPo. Lora32u4 memiliki dukungan bawaan untuk USB, hal ini memiliki keuntungan tetapi juga kerugian. Setelah reset, *serial port* akan terputus untuk sementara dan dialihkan ke mode pemrograman dan secara otomatis beralih kembali setelah beberapa detik. Terkadang hal ini dapat memberikan masalah saat mengupload atau saat menggunakan Serial Monitor. DIO1 pada Lora32u4 II v1.4 ini terletak pada pin 6, berbeda dengan versi yang lain.

g. Sensor Ultrasonic

Sensor ultrasonic bekerja dengan memanfaatkan gelombang ultrasonic. Ultrasonic merupakan gelombang yang frekuensinya berada diatas gelombang suara. Sensor ultrasonic yang digunakan dalam penelitian ini yakni model SRF05. Sensor ini bekerja dengan menggunakan sebuah *transmitter* yang memancarkan gelombang ultrasonic dan sebuah *receiver* yang menerima kembali gelombang ultrasonic. Penghitungan data diukur dari kecepatan gelombang dikalikan waktu tempuh dibagi 2. Sensor ultrasonic SRF05 ini bekerja pada tegangan DC 5 Volt, beban arus sebesar 30 mA sampai dengan 50 mA, dengan gelombang frekuensi 40 Khz. Sensor ini mampu membaca jarak antara 3 cm sampai 400 cm. Gambar 2.1 menunjukkan sensor ultrasonic SRF05 yang digunakan.



Gambar 2. 1 Sensor Ultrasonic SRF05

h. Sensor Waterflow

Merupakan sensor yang digunakan untuk mengukur arus air yang mengalir. Sensor ini memiliki *pinwheel* didalamnya, yang terhubung dengan *integrated magnetic hall effect sensor* yang mengeluarkan *electrical pulse* untuk setiap kali putaran. Dengan menghitung *pulse* dari keluaran sensor, maka aliran air dapat diperkirakan.

Model sensor waterflow yang digunakan untuk penelitian ini ialah YF-S201. Model sensor ini bekerja pada tegangan 5 sampai 18V DC, dengan output 5V. Sensor ini dapat bekerja dengan kecepatan aliran hingga 30 liter/menit, dengan temperatur -25 sampai 80°C dan kelembaban 35% - 80% RH. Akurasi yang dimiliki sensor ini berkisar 10%. Maksimum tekanan air yang dapat diterima adalah 2 MPa. Gambar 2.2 menunjukkan sensor waterflow YF-S201 yang digunakan untuk penelitian ini.



Gambar 2. 2 Sensor Waterflow YF-S201

2.3. Sistem Peringatan Dini Banjir

Sistem Peringatan Dini atau *Early Warning System* banjir merupakan serangkaian proses yang dijalankan untuk melakukan deteksi terhadap ancaman bencana banjir. Beberapa parameter digunakan oleh sistem untuk mendeteksi akan terjadinya sebuah banjir. Menurut (Ginting and M. Putuhena, 2014), Sistem peringatan dini banjir dibangun untuk meminimalkan kerugian yang disebabkan oleh banjir. Dengan memanfaatkan teknologi yang ada, sistem peringatan dini dibuat.

Dalam mengatasi bencana banjir terdapat beberapa langkah yang dapat dilakukan mulai dari pencegahan agar tidak terjadi banjir. Sebelum bencana banjir terjadi, terdapat berbagai hal yang dapat dijadikan acuan apabila akan terjadi banjir. Seperti diantaranya hujan deras yang berkepanjangan dan naiknya permukaan air. Dengan menggunakan acuan-acuan tersebut, dapat dibangun sebuah sistem peringatan yang dapat memberikan informasi mengenai tanda-tanda banjir yang akan datang, sehingga tindakan pencegahan lebih lanjut dapat dilakukan dengan baik.

Dengan mengimplementasikan sistem peringatan dini banjir terutama pada daerah-daerah yang rawan terkena banjir, maka dampak yang dapat ditimbulkan akibat banjir dapat diminimalisir.

2.4. Teknologi LoRa

Teknologi LoRa merupakan teknologi jaringan komunikasi wireless yang bermanfaat untuk pengembangan *Internet of Things (IoT)*. Teknologi LoRa memiliki kelebihan apabila dibandingkan dengan teknologi komunikasi nirkabel yang lain. LoRa memiliki kelebihan dengan konsumsi dayanya yang rendah tetapi dengan jarak jangkauan yang jauh. Teknologi jenis ini sangat cocok digunakan untuk penerapan perangkat-perangkat yang didesain untuk keperluan *Internet of Things (IoT)*.

Dengan menggunakan teknologi LoRa, implementasi *IoT* dapat diperluas. Penerapan *IoT* yang semakin besar memberikan manfaat kepada masyarakat juga pemerintah atau pengelola daerah. Dengan menggunakannya secara benar, teknologi ini dapat mengatasi berbagai hal dalam berbagai bidang. Sistem peringatan dini soal bencana sampai dengan penciptaan *smart city* pun dapat dilakukan dengan teknologi ini.

Perkembangan dunia yang semakin digital membuat kebutuhan penerapan *Internet of Things (IoT)* semakin besar, hal ini membuat teknologi LoRa menjadi pilihan yang tepat digunakan untuk mengembangkan sistem perangkat pintar.

BAB III

ANALISIS DAN RANCANGAN SISTEM

3.1. Alat Dan Bahan

Alat dan Bahan yang digunakan dalam penembangan sistem ini meliputi hardware serta software seperti dijabarkan dibawah :

Hardware :

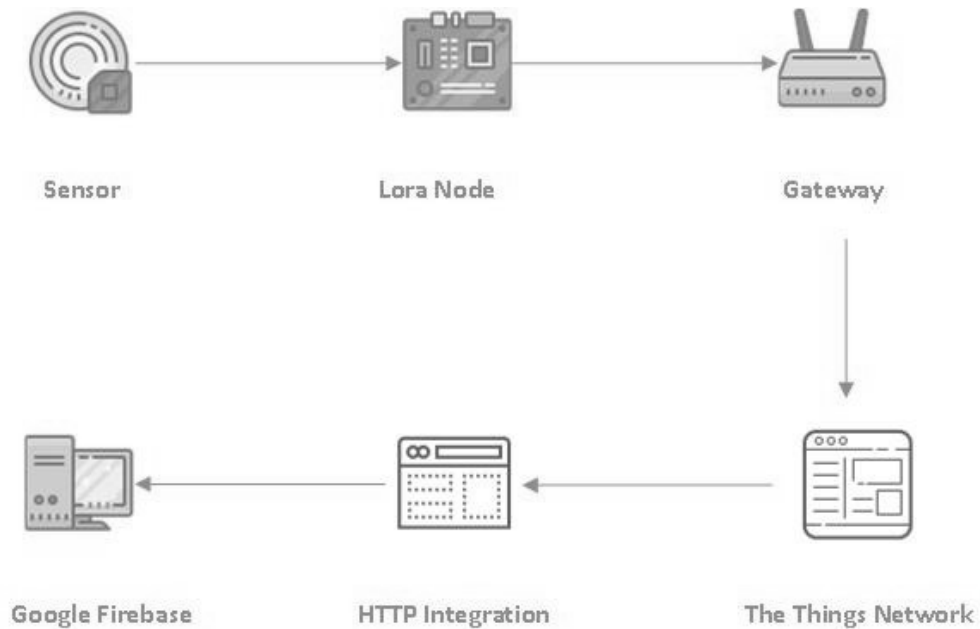
- Board BSFrance Lora32u4 II v1.4
- Gateway LG02
- Sensor Ultrasonic SRF05
- Sensor Waterflow YF-S201
- Kabel USB
- Laptop Ram 2GB

Software :

- Arduino IDE
- Google Chrome
- The Things Network
- Google Firebase

3.2. Alur Kerja Sistem

Sistem pendeteksian banjir ini bekerja dengan alur seperti pada gambar 3.1 dibawah :



Gambar 3. 1 Alur Kerja Sistem

Penjelasan mengenai alur kerja sistem dari Gambar 3.1 dijelaskan sebagai berikut :

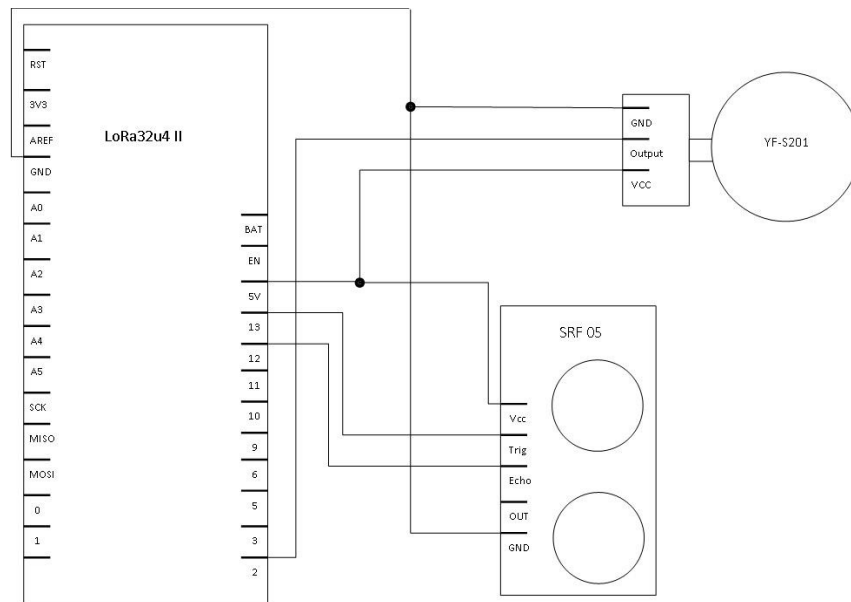
- Sensor dan LoRa Node bekerja dengan membaca data jarak ketinggian (*ultrasonic*) serta arus air (*waterflow*).
- Data sensor yang sudah tersimpan di Node lalu dikirimkan ke Gateway, LoRa Node mengirimkan *request* ke Gateway, setelah menerima *request*, gateway mengirimkan *reply* ke LoRa Node. Selanjutnya data sensor dapat dikirimkan ke Gateway.
- Gateway yang sudah dikonfigurasi dan terhubung dengan TTN (*The Things Network*) kemudian meneruskan data sensor yang diterima ke TTN.
- Data yang tersimpan sementara di TTN kemudian dihubungkan dengan layanan HTTP Integration, HTTP Integration digunakan untuk mengirimkan data yang tersimpan di TTN ke Google Firebase,

- e. Selanjutnya setelah konfigurasi HTTP Integration selesai dengan benar, data dikirimkan dan disimpan ke Google Firebase.

3.3. Desain Skematik

Perancangan perangkat keras berupa desain skematik dibuat untuk memudahkan pengguna dalam mendokumentasikan rancangan yang telah dibuat.

Desain skematik dari perangkat keras yang digunakan untuk pengembangan sistem ditunjukkan oleh Gambar 3.2.



Gambar 3. 2 Desain Skematik

Perancangan desain skematik diatas meliputi node dan sensor yang digunakan. *Board* yang digunakan merupakan LoRa32u4 II, dengan sensor ultrasonic SRF05 dan sensor arus air YF-S201. Pada rancangan diatas, penggunaan *ground* serta *vcc* pada kedua sensor digabung menjadi satu.

BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi

4.1.1. Konfigurasi Gateway

Konfigurasi Gateway dilakukan agar gateway dapat menerima dan meneruskan data, yang kemudian data dikirim lagi ke platform IoT. Pada pengembangan sistem ini digunakan Gateway Dragino dengan model LG02.

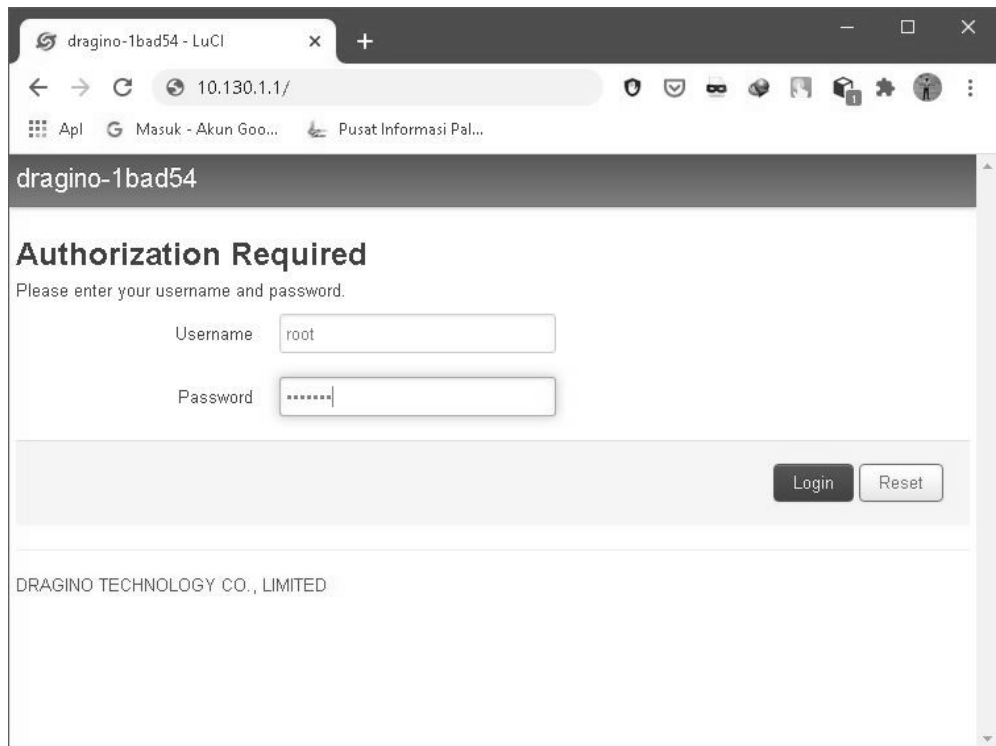
Diperlukan beberapa langkah agar Gateway dapat digunakan, berikut merupakan langkah-langkah konfigurasi Gateway :

a. Login Dragino.

Langkah pertama yakni login kedalam *web UI* Dragino. Setelah pemasangan perangkat gateway selesai, langkah selanjutnya ialah menyambungkan perangkat Gateway ke laptop dengan menggunakan kabel lan.

Setelah koneksi lan terhubung, langkah selanjutnya yakni membuka browser, kemudian masukkan alamat <http://10.130.1.1> pada search bar. Selanjutnya akan muncul halaman login Dragino, kemudian login dengan password “dragino”.

Tampilan halaman login Dragino ditunjukkan oleh Gambar 4.1.



Gambar 4. 1 Halaman *Login* Dragino

b. Konfigurasi LoRaWan Gateway.

Setelah masuk ke halaman utama, langkah selanjutnya adalah melakukan konfigurasi LoRaWan Gateway untuk dapat terhubung dengan platform The Things Network. Konfigurasi ini dapat dilakukan dengan membuka halaman *Service* → *LoRaWan Gateway*.

Tampilan halaman LoRaWan Gateway ditunjukkan oleh Gambar 4.2.

The screenshot shows a web browser window with the address bar displaying "10.130.1.1/cgi-bin/luci/admin/gateway/gateway". The page title is "dragino-1bad54" and the navigation bar includes links for Status, System, Network, Service, and Logout. The main heading is "LoRa Gateway Settings" with a subtitle "Configuration to communicate with LoRa devices and LoRaWAN server". Below this is the "LoRaWAN Server Settings" section, which contains the following fields:

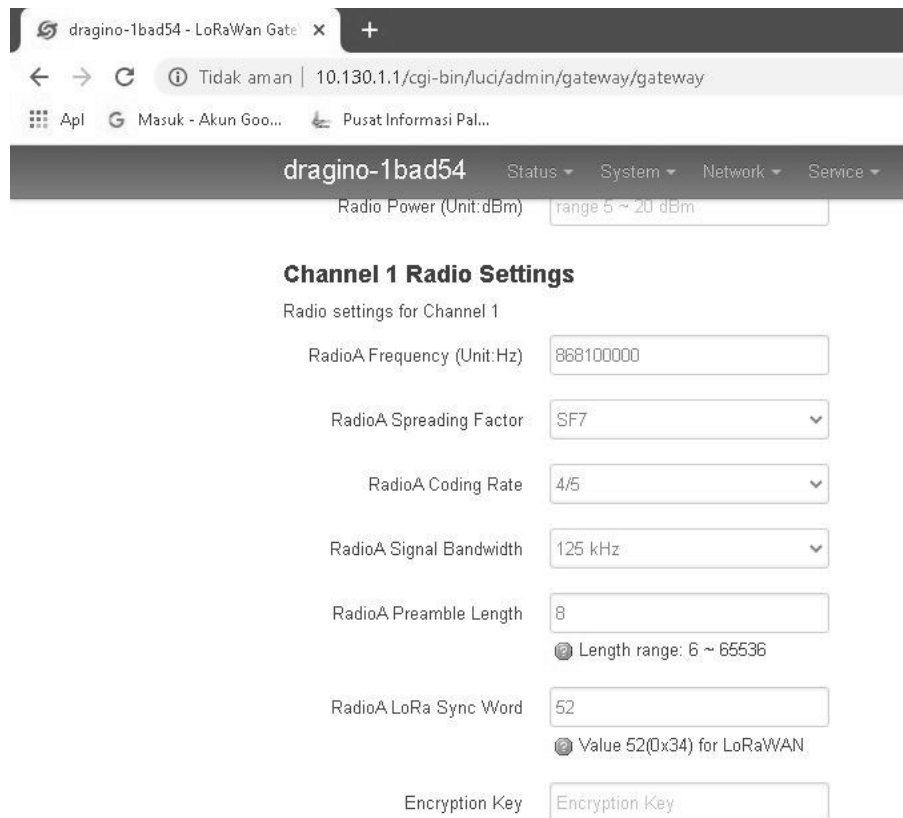
Field	Value
IoT Service	LoRaWan/RAW forwarder
Debug Level	No debug
Service Provider	The Things Network
Server Address	ttn-router-asia-se
Server Port	1700
Gateway ID	a840411bad540311
Mail Address	nanang.my@staff.uns.ac.id
Latitude	22.73
Longitude	114.23
RadioMode	A for RX, B for TX

Gambar 4.2 Konfigurasi *LoRaWan Server*

Gambar 4.2 menunjukkan konfigurasi *LoRaWan Server Settings*. Konfigurasi dilakukan dengan memilih *LoRaWan/Raw forwarder* pada *IoT Service*, *No Debug* pada *Debug Level*, *The Things Network* pada *Service Provider*, *ttn-router-asia-se* pada *Server Address*, serta mengisi opsi *Mail Address* yang tertera. Kolom *Gateway ID* digunakan untuk dapat tersambung dengan *platform IoT*.

Langkah selanjutnya ialah mengisi *Channel 1 Radio Settings*. Konfigurasi dilakukan dengan mengisi frekuensi radio A sebesar 868100000 atau frekuensi 868 MHz, kemudian dipilih SF7 untuk *RadioA Spreading Factor*.

Konfigurasi *Channel 1 Radio Settings* ditunjukkan oleh Gambar 4.3.



The screenshot shows a web browser window with the address bar displaying "10.130.1.1/cgi-bin/luci/admin/gateway/gateway". The page title is "dragino-1bad54". Below the title bar, there are tabs for "Status", "System", "Network", and "Service". The "Radio Power (Unit: dBm)" field is set to "range 5 ~ 20 dBm". The main section is titled "Channel 1 Radio Settings" and contains the following fields:

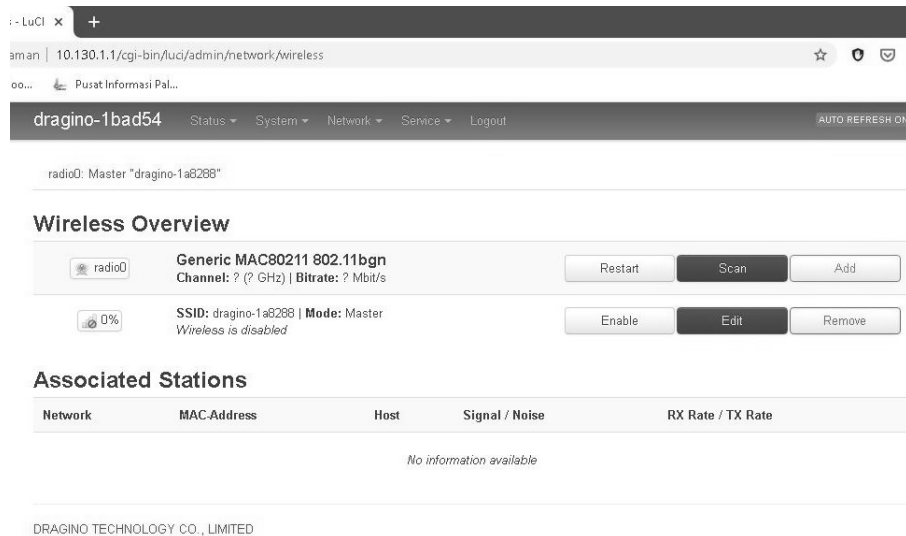
- Radio settings for Channel 1
- RadioA Frequency (Unit: Hz): 868100000
- RadioA Spreading Factor: SF7 (dropdown menu)
- RadioA Coding Rate: 4/5 (dropdown menu)
- RadioA Signal Bandwidth: 125 kHz (dropdown menu)
- RadioA Preamble Length: 8 (dropdown menu) with a note "Length range: 6 ~ 65536"
- RadioA LoRa Sync Word: 52 (dropdown menu) with a note "Value 52(0x34) for LoRaWAN"
- Encryption Key: Encryption Key (text input field)

Gambar 4. 3 Konfigurasi *Channel 1 Radio*

c. Konfigurasi Wireless Network.

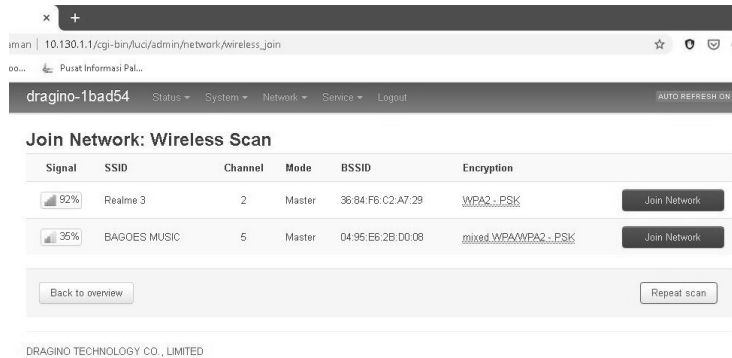
Setelah konfigurasi *LoRaWan Gateway* selesai dilakukan, langkah selanjutnya adalah melakukan konfigurasi *wireless network* agar Gateway online dan dapat terhubung dengan platform.

Konfigurasi wireless network dapat dilakukan dengan memilih menu *Network* → *Wireless*. Konfigurasi dari *wireless network* ditunjukkan pada Gambar 4.4.



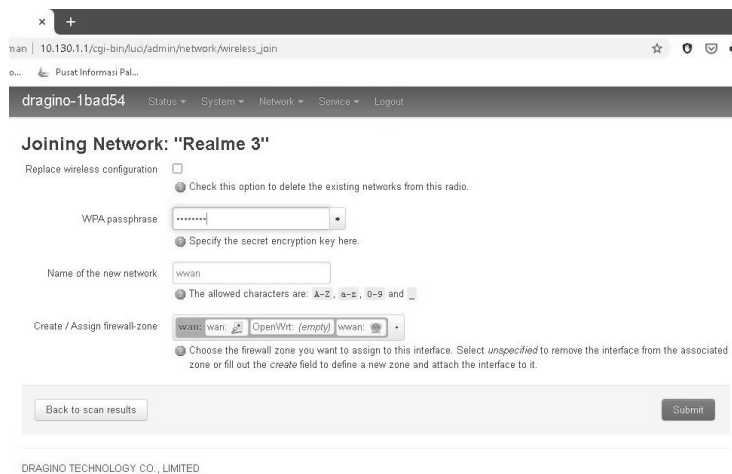
Gambar 4. 4 Tampilan Jaringan *Wireless*

Pada halaman *wireless network* seperti pada Gambar 4.4, pilih “*Scan*” pada “radio0”. Selanjutnya akan ditampilkan jaringan wireless yang ada seperti pada Gambar 4.5.



Gambar 4.5 Halaman *Join Network* Dragino

Pada Gambar 4.5 diatas pilih jaringan wireless yang akan disambungkan, kemudian pilih “Join Network”.



Gambar 4.6 Konfigurasi Jaringan *Wireless*

Seperti terlihat pada Gambar 4.6, pada jaringan yang akan disambungkan isikan *WPA passphrase*, setelah itu klik *Submit*. Kemudian akan muncul informasi lebih lanjut mengenai jaringan yang sudah dipilih, selanjutnya klik “*Save & Apply*”. Gambar 4.7 menunjukkan tampilan *Wireless Overview*.

dragino-1bad54 Status System Network Service Logout UNSAVED CHANGES: 12 AUTO REFRESH ON

Operating frequency Mode Channel Width
N 9 (2452 MHz) 20 MHz

Transmit Power auto dBm

Interface Configuration

General Setup Wireless Security Advanced Settings

Mode Client

ESSID Realme 3

BSSID EE:63:00:5B:17:76

Network wwan

Choose the network(s) you want to attach to this wireless interface or fill out the create field to define a new network.

Back to Overview Save & Apply Save Reset

DRAGINO TECHNOLOGY CO., LIMITED

Gambar 4. 7 Wireless Overview

4.1.2. Konfigurasi *The Things Network*

Konfigurasi platform ini dilakukan untuk menyimpan data dari gateway secara sementara untuk selanjutnya data diteruskan ke layanan database. Konfigurasi pada TTN (*The Things Network*) mencakup konfigurasi Gateway, konfigurasi *Application* dan *Device* serta konfigurasi *Integration*.

Berikut dibawah merupakan langkah-langkah pada konfigurasi platform *The Things Network* :

a. Konfigurasi Gateway.

Setelah membuat akun dan login ke dalam *The Things Network*, pilih menu *Gateway* selanjutnya pilih *Register Gateway*. Lalu akan ditampilkan form registrasi gateway seperti pada Gambar 4.8.

The screenshot shows the 'REGISTER GATEWAY' form in the The Things Network console. The form is titled 'REGISTER GATEWAY' and contains several sections:

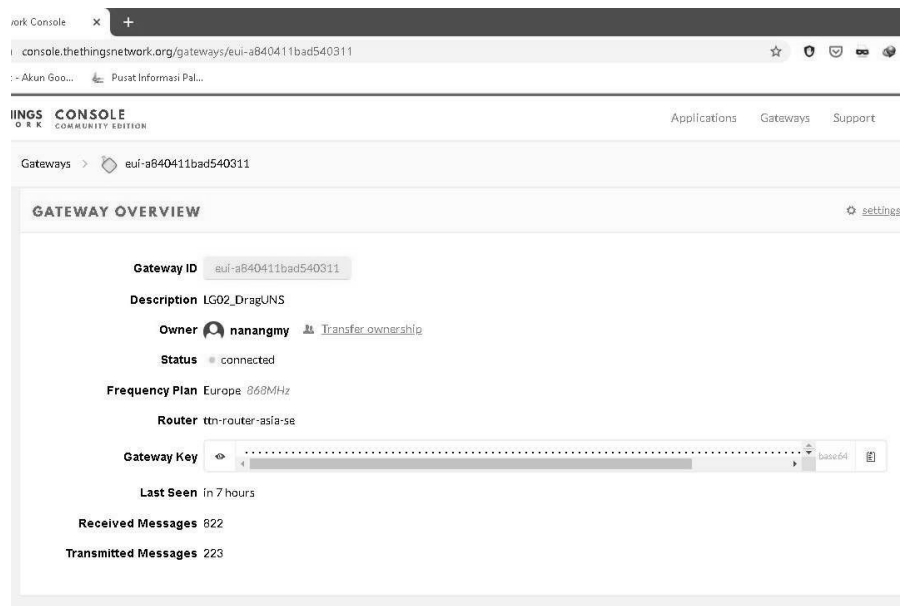
- Gateway ID**: A text input field with a placeholder 'A unique, human-readable identifier for your gateway. It can be anything so be creative!'.
- I'm using the legacy packet forwarder**: A checkbox with a label 'Select this if you are using the legacy Semtech packet forwarder'.
- Description**: A text input field with a placeholder 'A human-readable description of the gateway'.
- Frequency Plan**: A dropdown menu with a placeholder 'The frequency plan this gateway will use' and a value 'no selection'.
- Router**: A text input field with a placeholder 'The router this gateway will connect to. To reduce latency, pick a router that is in a region which is close to the location of the gateway'.
- Location**: A map of Jakarta with a placeholder 'The exact location of your gateway. This will be used if your gateway cannot determine its location by itself. Set a location by clicking on the map.' The map shows various districts like Jakarta Barat, Jakarta Utara, Jakarta, and Jakarta Timur.
- Antenna Placement**: Two radio buttons labeled 'indoor' and 'outdoor'.

At the bottom right of the form, there are two buttons: 'Cancel' and 'Register Gateway'.

Gambar 4. 8 Halaman Registrasi Gateway

Sesuai dengan form seperti pada gambar 4.8, isikan *Gateway ID* sesuai dengan *Gateway ID* yang ada pada konfigurasi perangkat gateway sebelumnya, kemudian *checkbox* pada pilihan *I'm using the legacy packet forwarder*. Pada *Description* diisikan “LG02_DragUNS”, lalu pada pilihan *Frequency Plan* dipilih “Europe 868MHz”, pada pilihan *Router* dipilih “ttn-router-asia-se”, kemudian dipilih *Location* yang sesuai dan dipilih “Indoor” sebagai *Antenna Placement*. Setelah selesai klik “Register Gateway”.

Setelah konfigurasi selesai, pada halaman gateway akan ditampilkan informasi mengenai gateway yang sudah ditambahkan seperti pada Gambar 4.9.



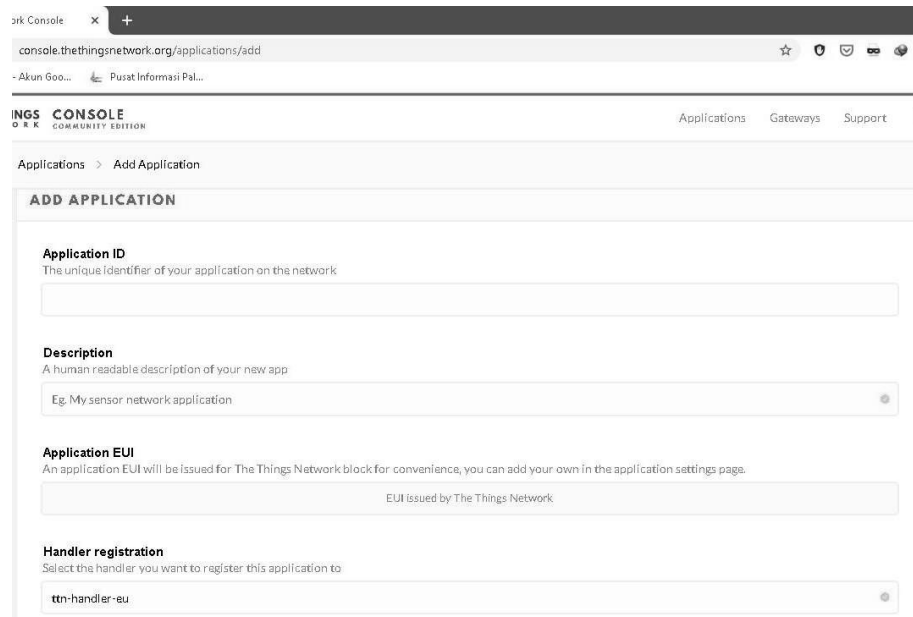
Gambar 4.9 Halaman Gateway Overview

Status pada Gambar 4.9 diatas menunjukkan apakah gateway tersambung atau tidak dengan platform.

b. Konfigurasi Application & Device.

Data yang diterima pada platform disimpan pada layanan *application* di platform. Didalam *application* ditambahkan sebuah *device* yang spesifik menerima data dari *node* yang ada. berikut merupakan langkah-langkah konfigurasi *application* dan *device* pada platform.

Penambahan *application* pada platform dapat dilakukan dengan memilih menu *Application* → *Add Application*. Kemudian akan ditampilkan form penambahan *application*.

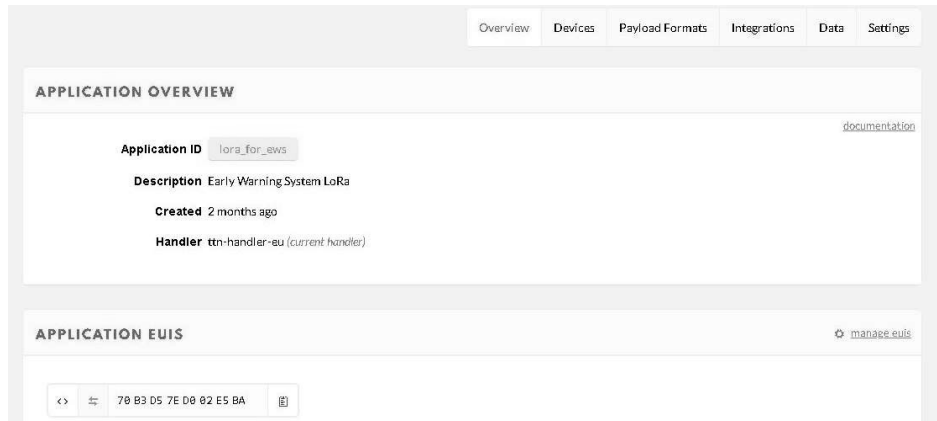


The screenshot shows a web browser window with the URL `console.thethingsnetwork.org/applications/add`. The page title is 'INES CONSOLE COMMUNITY EDITION'. The breadcrumb navigation shows 'Applications > Add Application'. The main heading is 'ADD APPLICATION'. The form contains the following sections:

- Application ID**: A text input field with the placeholder 'The unique identifier of your application on the network'.
- Description**: A text input field with the placeholder 'A human readable description of your new app' and an example 'Eg. My sensor network application'.
- Application EUI**: A text input field with the placeholder 'An application EUI will be issued for The Things Network block for convenience, you can add your own in the application settings page.' and a default value 'EUI issued by The Things Network'.
- Handler registration**: A dropdown menu with the placeholder 'Select the handler you want to register this application to' and a selected option 'ttn-handler-eu'.

Gambar 4. 10 Halaman *Add Application*

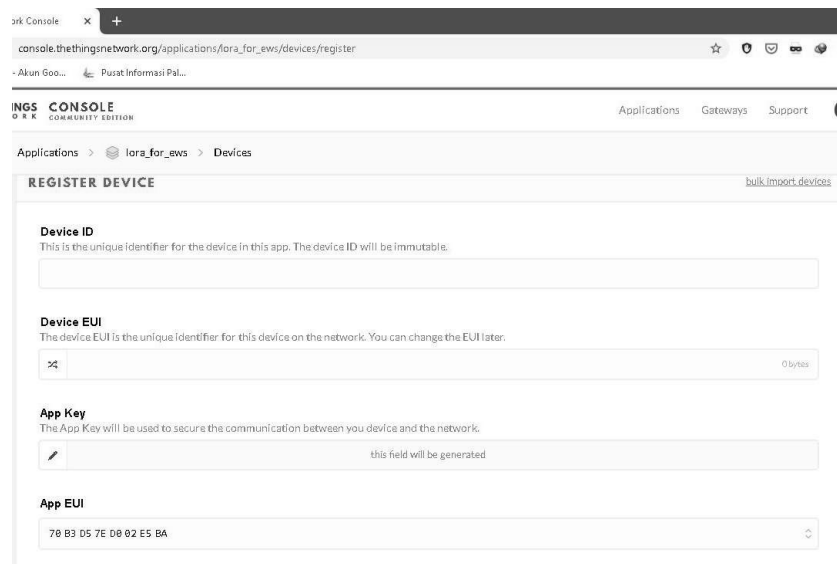
Pada menu *Add Application* seperti yang terlihat pada gambar 4.10, user perlu mengisi *Application ID*, *Description* dan memilih *Handler registration*, untuk *Application EUI* akan diisi secara otomatis oleh platform, setelah itu klik submit.



Gambar 4. 11 Application Overview

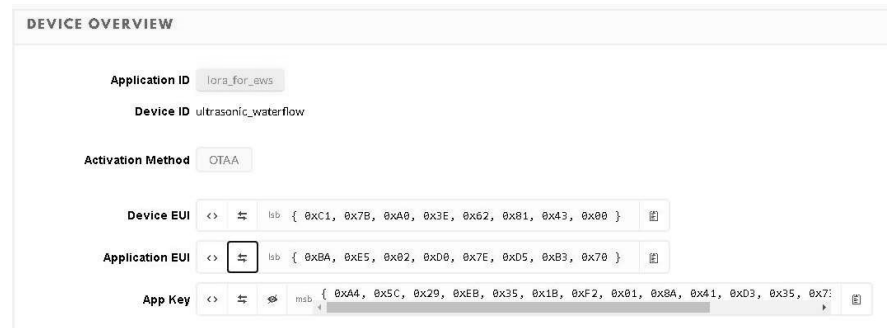
Setelah pengisian application selesai, akan ditampilkan *application overview* seperti pada Gambar 4.11. Langkah selanjutnya adalah menambahkan *device*. Untuk menambahkan *device* dapat dilakukan dengan memilih menu *device* pada *application* lalu *register device*.

Pada form penambahan *device*, user dapat mengisi *Device ID* lalu klik submit. Gambar 4.12 menampilkan penambahan *device* :



Gambar 4. 12 Halaman Registrasi Device

Setelah pengisian form penambahan *device* selesai, akan didapatkan *Device EUI*, *Application EUI* dan *App Key* yang nantinya akan digunakan pada *source code* untuk node agar dapat dikenali. Gambar 4.13 merupakan *Device Overview* :



Gambar 4. 13 Tampilan *Device EUI*, *App EUI* dan *App Key*

c. Konfigurasi Integration.

Penggunaan *Integration* dimaksudkan agar data yang telah diterima dapat dikirimkan ke layanan basisdata dalam hal ini Google Firebase. Untuk layanan *Integration* yang dipilih untuk mengirimkan data ke Firebase ialah *HTTP Integration*. Berikut merupakan langkah-langkah penambahan *Integration* pada platform.

Untuk menambahkan layanan *Integration* pada TTN, dapat dilakukan pada menu *Application* → *Integrations*, lalu *add integration*. Kemudian akan ditampilkan form seperti pada Gambar 4.14 dibawah :

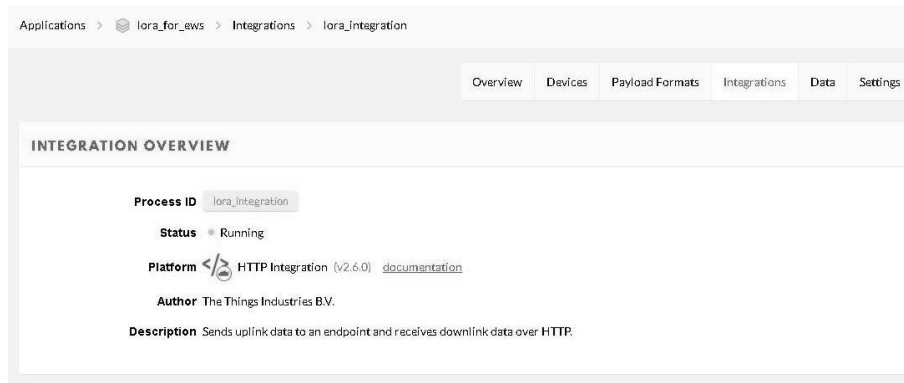
The screenshot shows the 'Create HTTP Integration' page in the Things Network Console. The breadcrumb navigation is 'Applications > lora_for_ews > Integrations'. The form contains the following fields:

- Process ID**: A text input field with the description 'The unique identifier of the new integration process'.
- Access Key**: A dropdown menu with the description 'The access key used for downlink'. The selected option is 'default key', with other visible options being 'devices' and 'messages'.
- URL**: A text input field with the description 'The URL of the endpoint'.
- Method**: A dropdown menu with the description 'The HTTP method to use'. The selected option is 'POST'.

Gambar 4. 14 Halamana Penambahan *HTTP Integration*

Process ID digunakan sebagai identitas *integration* yang ada, untuk *Access Key* dipilih *default key*, dan *Post* sebagai *Method*. *URL* yang digunakan didapatkan dari layanan database Firebase yang telah dibuat, pada percobaan ini *URL* yang digunakan ialah `lora-db.firebaseio.com/lora/sensor.json`.

Setelah penambahan selesai dan dilakukan dengan benar, maka pada halaman *Integration* akan ditampilkan seperti Gambar 4.15 berikut :



Gambar 4. 15 Integration Overview

4.1.3. Implementasi Source Code

Langkah-langkah implementasi source code dibagi menjadi 2 yakni konfigurasi library dan implementasi source code.

Source code yang digunakan dalam pengembangan sistem ini menggunakan beberapa library yang kemudian digabung menjadi satu. *Library* yang digunakan yakni arduino-lmic-master. Sebelum dapat digunakan, dibutuhkan konfigurasi yang sesuai dengan spesifikasi node. Berikut dibawah merupakan konfigurasi library arduino-lmic-master.

a. Konfigurasi file config.

Konfigurasi yang pada library dilakukan pada file arduino-lmic-master/src/lmic/config.h. Konfigurasi yang dibutuhkan yakni hanya dengan menyesuaikan frekuensi yang hendak digunakan. Konfigurasi ini dapat dilihat pada Gambar 4.16.

```

1  #ifndef _lmic_config_h
2  #define _lmic_config_h
3
4  // In the original LMIC code, these config values were defined on the
5  // gcc commandline. Since Arduino does not allow easily modifying the
6  // compiler commandline, use this file instead.
7
8
9  #define CFG_eu868 1
10 // #define CFG_us915 1
11 // #define CFG_as923 1
12 // #define CFG_as923 1
13 // #define CFG_in866 1
14
15 #define LG02_LG01 1
16
17 //US915: DR_SF10=0, DR_SF9=1, DR_SF8=2, DR_SF7=3, DR_SF8C=4
18 //      DR_SF12CR=8, DR_SF11CR=9, DR_SF10CR=10, DR_SF9CR=11, DR_SF8CR=12, DR_SF7CR
19 #if defined(CFG_us915) && defined(LG02_LG01)
20 // CFG_us915 || CFG_as923
21 #define LG02_UPFREQ 902320000
22 #define LG02_DNWFREQ 923300000
23 #define LG02_RXSF 3 // DR_SF7 For LG01/LG02 Tx
24 #define LG02_TXSF 8 // DR_SF12CR For LG02/LG02 Rx
25 #elif defined(CFG_eu868) && defined(LG02_LG01)

```

Gambar 4. 16 Tampilan Konfigurasi File *Config*

Seperti yang terlihat pada gambar 4.16, frekuensi yang digunakan yakni 868.

Setelah melakukan konfigurasi pada library, langkah selanjutnya ialah implementasi *source code*. *Source code* yang digunakan pada pengembangan ini melibatkan *open source code* dari sensor ultrasonic, sensor waterflow, serta *source code* yang terdapat pada library, kemudian ketiga *source code* tersebut digabung menjadi satu.

Beberapa tampilan *source code* yang digunakan pada pengembangan sistem ini terlihat seperti pada Gambar 4.17 dibawah.

```

TTN-OTAA-LMIC_Ultrasonic_Waterflow

#include <lmic.h>
#include <hal/hal.h>
#include <SPI.h>

// Waterflow -----
#define FLOWSENSORPIN 2
// count how many pulses!
volatile uint16_t pulses = 0;
volatile uint16_t pulses2 = 0;
// track the state of the pulse pin
volatile uint8_t lastflowpinstate;
// you can try to keep time of how long it is between pulses
volatile uint32_t lastflowratetimer = 0;
// and use that to calculate a flow rate
volatile float flowrate;
// Liters/Second for send
float liters_s;
// Liters/Minute for send
float liters_m;
// Liters/Second for send
float liters_hour;
//-----

// Ultrasonic -----
const unsigned int TRIG_PIN=13; // --
const unsigned int ECHO_PIN=12; // --
int depth = 15;

```

Gambar 4. 17 Sketch : pendefinisian variabel

Pada Gambar 4.17 diatas ditampilkan pendefinisian variabel-variabel untuk masing-masing sensor beserta pin yang digunakan.

```

// This EUI must be in little-endian format, so least-significant-byte
// first. When copying an EUI from ttnctl output, this means to reverse
// the bytes. For TTN issued EUIs the last bytes should be 0xD5, 0xB3,
// 0x70.
static const ul_t PROGMEM APPEUI[8]= { 0xBA, 0xE5, 0x02, 0xD0, 0x7E, 0xD5, 0xB3, 0x70 };
void os_getArtEui (ul_t* buf) { memcpy_P(buf, APPEUI, 8);}

// This should also be in little endian format, see above.
static const ul_t PROGMEM DEVEUI[8]= { 0xC1, 0x7B, 0xA0, 0x3E, 0x62, 0x81, 0x43, 0x00 };
void os_getDevEui (ul_t* buf) { memcpy_P(buf, DEVEUI, 8);}

// This key should be in big endian format (or, since it is not really a
// number but a block of memory, endianness does not really apply). In
// practice, a key taken from ttnctl can be copied as-is.
// The key shown here is the semtech default key.
static const ul_t PROGMEM APPKEY[16] = { 0xA4, 0x5C, 0x29, 0xEB, 0x35, 0x1B, 0xF2, 0x01, 0x8A,
void os_getDevKey (ul_t* buf) { memcpy_P(buf, APPKEY, 16);}

//static uint8_t mydata[3];
static osjob_t sendjob;

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).
const unsigned TX_INTERVAL = 900;

```

Gambar 4. 18 Sketch : *Device EUI, Application EUI dan App Key*

Gambar 4.18 diatas menunjuk *Device EUI, Application EUI* dan *App Key*, berikut interval waktu yang digunakan untuk pengiriman yakni 30 detik.

```

void loop() {
  os_runloop_once();

  // Waterflow -----
  float liters = pulses;
  liters /= 7.5;
  liters /= 60.0;
  float liters_h = pulses2;
  liters_h *= 60.0;
  liters_h /= 7.5;
  liters_s = liters_h/3600;
  liters_m = liters_h/60;
  liters_hour = liters_h;
  pulses2 = 0;
  delay(100);
  //-----
}

```

Gambar 4. 19 Sketch : *fungsi loop sensor waterflow*

Gambar 4.19 menunjukkan *function loop()*. Pada *function loop()* disisipkan source code yang digunakan untuk sensor waterflow.

```
void do_send(osjob_t* j){
    // Ultrasonic -----
    digitalWrite(TRIG_PIN, LOW);
    delayMicroseconds(2);
    digitalWrite(TRIG_PIN, HIGH);
    delayMicroseconds(10);
    digitalWrite(TRIG_PIN, LOW);
    const unsigned long duration= pulseIn(ECHO_PIN, HIGH);
    int distance = duration/29/2;
    int water_level = depth-distance;
    if(duration==0){
        Serial.println("Warning: no pulse from Ultrasonic");
    }
    else{
        Serial.print("Jarak Objek Terdekat:");
        Serial.print(distance);
        Serial.println(" cm");
    }
    //-----

    // Waterflow -----
    Serial.print("liters/second : ");Serial.print(liters_s); Serial.println(" l/s");
    Serial.print("liters/minute : ");Serial.print(liters_m); Serial.println(" l/m");
    Serial.print("liters/hour : ");Serial.print(liters_hour); Serial.println(" l/h");
    //-----

    String data = (String)water_level+","+ (String)liters_m;
    Serial.println(data);

    char mydata[16];
    data.toCharArray(mydata,16);
    Serial.print("mydata : ");Serial.println(mydata);
    |
    // Check if there is not a current TX/RX job running
    if (LMIC.opmode & OP_TXRXPEND) {
        Serial.println(F("OP_TXRXPEND, not sending"));
    } else {
        // Prepare upstream data transmission at the next possible time.
        LMIC_setTxData2(1, mydata, strlen(mydata), 0);
        Serial.println(F("Packet queued"));
    }

    // Next TX is scheduled after TX_COMPLETE event.
}
```

Gambar 4. 20 Sketch : Fungsi pengiriman data

Function *do_send()* seperti terlihat pada Gambar 4.20 diatas, digunakan untuk mengirimkan data sensor yang sudah ada ke platform.

Pada *function* ini juga terdapat *source code* yang digunakan untuk sensor ultrasonic. Pada gambar diatas dapat dilihat bahwa masing-masing data sensor digabungkan menjadi satu kemudian dimasukkan ke-variabel *mydata[16]* setelah itu dikirim. Keseluruhan *source code* pengembangan sistem ini dapat dilihat di <https://gitd3ti.mipa.uns.ac.id/wahyukris/ews-lora>.

4.2. Pengujian

4.2.1. Pengujian Sensor

Pengujian sensor dilakukan dengan menguji kesamaan data yang dikirimkan dan jeda waktu yang diperlukan untuk mengirimkan data sampai ke-Firebase.

Pengujian dilakukan secara bersamaan untuk sensor *ultrasonic* dan arus air. Pengujian sensor ultrasonic dilakukan dengan mengukur jarak yang ditangkap oleh sensor dengan jarak real yang ada. Sedangkan untuk pengujian sensor *waterflow* dilakukan dengan menguji arus air yang ada dengan jumlah air yang didapat dalam periode waktu 30 detik. Gambar 4.21 menunjukkan rangkaian *device* yang digunakan saat pengujian.



Gambar 4. 21 Rangkaian *device* untuk pengujian I

Gambar 4.21 diatas merupakan proses pengujian sensor yang dilakukan dengan menggunakan ember untuk menampung air dan selotip untuk menghitung jarak.

Sensor arus air yang digunakan terhubung dengan sebuah kran yang dapat diatur besar kecilnya arus yang dikeluarkan. Jumlah air kemudian dihitung secara manual berapa liter yang didapatkan.



Gambar 4. 22 Rangkaian *device* untuk pengujian II

Gambar 4.22 menunjukkan rangkaian device untuk pengujian. Proses pengujian sensor ultrasonic dilakukan secara manual dengan menggunakan penggaris untuk menghitung jarak *real* yang sudah diatur.

4.2.2. Hasil Pengujian

Setelah pengujian selesai dilakukan, diambil data sejumlah 5 untuk masing-masing sensor. Hasil yang dibandingkan pada pengujian ini diambil dari informasi yang terdapat pada *Serial Monitor* pada aplikasi *Arduino*, *Data Device* pada platform, dan data yang muncul pada *Firebase*.

Setelah diolah berikut ditampilkan data yang didapatkan dari pengujian yang sudah dilakukan. Tabel 4.1 menunjukkan data hasil pengujian sensor ultrasonic dan waterflow.

Tabel 4. 1 Data hasil pengujian

Data ke-	Data		
	Waterflow		Ultrasonic
	liter/detik	liter/jam	cm
1	0,00	16,00	9
2	0,01	32,00	17
3	0,02	56,00	37
4	0,03	104,00	55
5	0,05	176,00	82

Pada Tabel 4.1 diatas data yang ditampilkan pada masing-masing aplikasi yakni *Serial Monitor*, *Data Device*, dan database *Firebase* sudah sama sehingga hanya ditampilkan 1 untuk masing-masing data.

Perbandingan waktu yang didapatkan pada setiap aplikasi ditampilkan dalam Tabel 4.2.

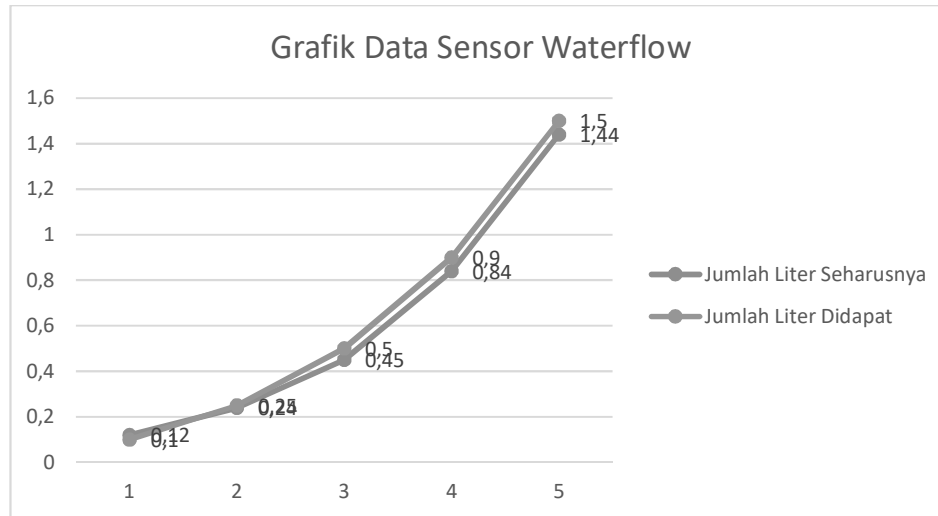
Tabel 4. 2 Perbedaan waktu antar layanan

Data ke-	Time			Selisih Waktu
	Serial Monitor	TTN	Firebase	
	UTC +7	UTC +0	UTC +0	
1	16:25:56	09:25:56	09:25:56	0 Detik
2	02:30:00	19:29:59	19:29:59	1 Detik
3	02:49:58	19:49:58	19:49:58	0 Detik
4	02:58:34	19:58:34	19:58:34	0 Detik
5	03:11:56	20:11:56	20:11:56	0 Detik

Perbandingan data pada sensor waterflow ditampilkan dengan membandingkan jumlah liter seharusnya (didapatkan dari hitungan manual jumlah liter/detik) dengan jumlah liter yang didapatkan secara real. Tabel 4.3 dan Grafik 4.1 menunjukkan perbandingan data sensor waterflow.

Tabel 4. 3 Perbandingan Data Sensor Arus Air

Data ke-	Jml Liter	Jml Liter Didapat	Selisih (liter)
	Seharusnya	Dalam 30 Detik	
1	0,12	0,1	0.02
2	0,24	0,25	0.01
3	0,45	0,5	0.05
4	0,84	0,9	0.06
5	1,44	1,5	0.06

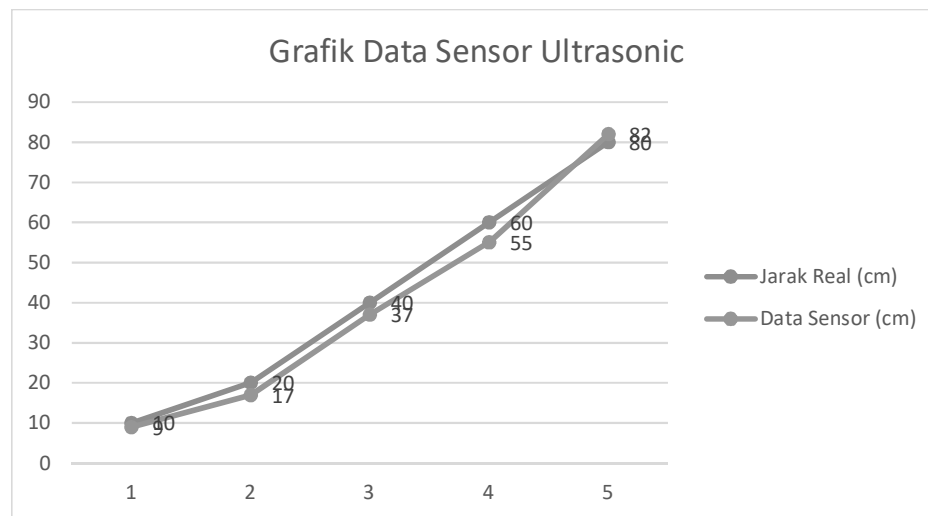


Grafik 4. 1 Data Sensor Arus Air

Perbandingan data pada sensor ultrasonic ditampilkan dengan membandingkan jarak data yang didapatkan oleh sensor dengan jarak real. Tabel 4.4 dan Grafik 4.2 menunjukkan perbandingan data sensor ultrasonic.

Tabel 4. 4 Perbandingan Data Sensor *Ultrasonic*

Data ke-	Jarak Real (cm)	Data Sensor (cm)	Selisih (cm)
1	10	9	1
2	20	17	3
3	40	37	3
4	60	55	5
5	80	82	2



Grafik 4. 2 Data Sensor *Ultrasonic*

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan penelitian yang sudah dilakukan seperti penjelasan diatas, maka dapat disimpulkan bahwa pengembangan *early warning system* banjir dengan teknologi LoRa menunjukkan hasil yang cukup baik untuk dapat menguji parameter yang ada. Pada penelitian ini juga dibuktikan bahwa pengiriman data yang dilakukan dengan menggunakan teknologi lora berjalan dengan baik.

Hasil pengujian yang dilakukan menunjukkan data yang diperoleh belum 100 persen akurat, akan tetapi secara keseluruhan proses yang dijalankan dari pengambilan data dari sensor sampai dengan pengiriman yang dilakukan berjalan dengan baik. Data-data yang dikumpulkan dan disimpan dalam basis data digunakan untuk mengembangkan aplikasi yang dapat memantau kondisi sungai secara *realtime*.

5.2. Saran

Pengembangan sistem peringatan dini untuk banjir dengan menggunakan teknologi LoRa ini tentunya tidak luput dari kekurangan dan memerlukan pengembangan lebih lanjut untuk dapat menyempurnakannya. Seperti halnya perangkat lunak yang digunakan yang terus mengalami pembaruan, sistem peringatan ini juga memerlukan pembaruan. Perangkat-perangkat yang digunakan juga dapat diperbarui untuk versi yang lebih baru sehingga data-data yang didapatkan lebih akurat. Pengambilan data yang dilakukan dapat diperbanyak untuk dapat menganalisa keakuratan dari sensor yang digunakan.

DAFTAR PUSTAKA

- Adin, D. T., Bhawiyuga, A. and Yahya, W. (2019) 'Sistem Monitoring Parameter Fisik Air Kolam Ikan menggunakan Jaringan Sensor Nirkabel berbasis Protokol LoRa', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3, pp. 5414–5420.
- Ginting, S. and M. Putuhena, W. (2014) 'SISTEM PERINGATAN DINI BANJIR JAKARTA JAKARTA-FLOOD EARLY WARNING SYSTEM (J-FEWS)', *Jurnal Sumber Daya Air*, 10(1), pp. 71–84. Available at: <http://journalsda.pusair-pu.go.id/index.php/JSDA/article/download/144/132>.
- Susanto, A. R., Bhawiyuga, A. and Amron, K. (2019) 'Implementasi Sistem Gateway Discovery pada Wireless Sensor Network (WSN) Berbasis Modul Komunikasi LoRa', *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, 3, pp. 2138–2145.