



**Analisis Performansi Pengiriman Data
Menggunakan Pendekatan QoS Metrics pada
Platform IoT Ubidots dan ThingSpeak, Studi
Kasus: *Monitoring* Aliran Air Rumah Pintar**

Tugas Akhir

**Oleh:
Nurul Khairani (4212001044)**

**Program Studi Teknik Mekatronika
Jurusan Teknik Elektro
Politeknik Negeri Batam
2024**

Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan dibawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : "Analisis Performansi Pengiriman Data Menggunakan Pendekatan QoS Metrics pada Platform IoT Ubidots dan ThingSpeak, Studi Kasus: Monitoring Aliran Air Rumah Pintar" adalah hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 16 Januari 2024



Nurul Khairani
NIM: 4212001044



Asri Arganing Hasanah
NIM: 4212001041

Lembar Pengesahan

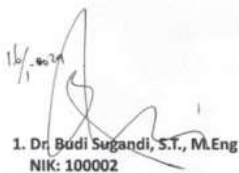
Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Terapan Teknik (S.Tr.T)
di
Politeknik Negeri Batam


Oleh:

Asri Arganing Hasanah (4212001041)
Nurul Khairani (4212001044)

Tanggal Sidang: 02 01, 2024

Disetujui oleh :


1. Dr. Budi Sugandi, S.T., M.Eng
NIK: 100002


1. Heru Wijanarko, S.T., M. Sc
NIK: 110070


2. Dr. Abdurahman Dwijotomo, S.ST., M.Sc
NIK: 122257


2. M. Naufal-Airlangga Diputra, S.Pd., M.P.h
NIK: 122281

Analisis Performansi Pengiriman Data Menggunakan Pendekatan *QoS Metrics* pada *Platform IoT Ubidots* dan *ThingSpeak*, Studi Kasus: *Monitoring Aliran Air Rumah Pintar*

Abstrak

Perkembangan teknologi IoT dan berbagai *platform*-nya yang memungkinkan *monitoring* jarak jauh secara *real-time*. Penelitian ini fokusnya membuat sistem *monitoring* aliran air pada rumah pintar berbasis IoT dengan sensor *water flow* YF-S201 dan *mikrokontroler* ESP32, serta membandingkan performansi pengiriman data *platform* IoT *Ubidots* dan *ThingSpeak* dengan pendekatan *QoS metric* dan penerapan model RNN-LSTM. Metodologi mencakup perancangan dan realisasi sistem *monitoring* aliran air, dilanjutkan dengan pengambilan data sensor *water flow*, pengiriman dan analisis data menggunakan *platform* IoT *Ubidots* dan *ThingSpeak*, serta penerapan RNN-LSTM untuk prediksi data sensor. Hasil penelitian menunjukkan bahwa sistem *monitoring* berhasil dirancang dan diimplementasikan dengan baik, data sensor *water flow* berhasil dikirim dan ditampilkan pada kedua *platform* IoT *Ubidots* dan *ThingSpeak* secara *real-time*. Namun, *Ubidots* lebih cepat dengan *delay* total 5 detik dengan tampilan *dashboard*-nya lebih informatif, dibandingkan *ThingSpeak* memiliki rata-rata *delay* 17 detik. Analisis *QoS* menunjukkan performansi *Ubidots* lebih unggul dibanding *ThingSpeak* pada parameter *jitter* dan *delay*, serta model RNN-LSTM mampu memprediksi data aktual serta data aplikasi sensor di masa depan dengan tingkat error yang kecil. Sehingga *Ubidots* dapat menjadi pilihan yang baik untuk pengiriman data sensor dan *QoS metrics*

Kata kunci: *Ubidots vs ThingSpeak, Quality of Service, RNN-LSTM*

Performance Analysis of Data Transmission Using QoS Metrics Approach on IoT Platforms Ubidots and ThingSpeak, Case Study: Smart Home Water Flow Monitoring

Abstract

The development of IoT technology and its various platforms allows real-time remote monitoring. This research focuses on creating a water flow monitoring system in an IoT-based smart home with a YF-S201 water flow sensor and an ESP32 microcontroller, and comparing the data transmission performance of the Ubidots and ThingSpeak IoT platforms with the QoS metric approach and the application of the RNN-LSTM model. The methodology includes the design and realization of a water flow monitoring system, followed by water flow sensor data collection, data transmission and analysis using the Ubidots and ThingSpeak IoT platforms, and the application of RNN-LSTM for sensor data prediction. The results showed that the monitoring system was successfully designed and implemented well, the water flow sensor data was successfully sent and displayed on both Ubidots and ThingSpeak IoT platforms in real-time. However, Ubidots is faster with a total delay of 5 seconds with a more informative dashboard display, compared to ThingSpeak which has an average delay of 17 seconds. QoS analysis shows Ubidots performance is superior to ThingSpeak on jitter and delay parameters, and the RNN-LSTM model is able to predict actual data and future sensor application data with a small error rate. So Ubidots can be a good choice for sensor data delivery and QoS metrics.

Keywords: Ubidots vs ThingSpeak, Quality of Service, RNN-LSTM

Kata Pengantar

Segala puji dan syukur penulis haturkan kehadiran Allah SWT. yang Maha Esa, atas segala rahmat, kasih, hidayah dan karunia-Nya. Shalawat serta salam semoga kita curahkan kepada baginda nabi Muhammad SAW. sehingga penulis dapat menyelesaikan tugas akhir dan laporan akhir yang berjudul “Analisis Performansi Pengiriman Data Menggunakan Pendekatan QoS *Metrics* pada Platform IoT *Ubidots* dan *ThingSpeak*, Studi Kasus: Monitoring Aliran Air Rumah Pintar”. Tugas akhir ini disusun untuk memenuhi salah satu syarat dalam memperoleh gelar Sarjana Terapan Teknik pada Jurusan Elektro, Program Studi Teknik Mekatronika, Politeknik Negeri Batam.

Penulis menyadari bahwa dalam penyusunan tugas akhir ini terwujud berkat bantuan, arahan, bimbingan, dan doa dari berbagai pihak. Penulis ingin menyampaikan rasa terima kasih yang sebesar - besarnya yang diberikan dalam setiap proses penyelesaian tugas akhir ini kepada:

1. Tuhan Yang Maha Esa atas segala rahmat, kasih, hidayah dan karunia-Nya.
2. Kedua orang tua penulis yang selalu memberikan doa restu, nasihat bimbingan serta dukungan moril dan materil yang tak henti-hentinya diberikan.
3. Kepada abang, kakak, dan adik penulis yang selalu memberikan dorongan, nasihat dan kesabaran dalam setiap langkah perjalanan penulis.
4. Bapak Uuf Brajawidagda, S.T., M.T., Ph.D selaku Direktur Politeknik Negeri Batam.
5. Bapak Dr. Budi Sugandi, S.T., M.Eng selaku Ketua Jurusan Elektro Politeknik Negeri Batam dan dosen penguji I penulis yang telah memberikan pertanyaan kritis, masukan dan saran dalam penulisan tugas akhir ini.
6. Bapak Indra Hardian Mulyadi, S.T., M.Eng., Ph.D selaku Ketua Program Studi D-IV Teknik Mekatronika.
7. Bapak Sumantri Kurniawan Risandriya, ST, MT selaku dosen wali penulis.
8. Bapak Heru Wijanarko, S.T., M.Sc selaku dosen pembimbing I yang selalu meluangkan waktu untuk memberi bimbingan, arahan, motivasi dan bantuan yang berharga kepada penulis agar selalu tetap semangat dalam menyelesaikan tugas akhir.
9. Bapak Muhammad Naufal Airlangga Diputra, S.Pd., M.P.H selaku dosen pembimbing II dan dosen pengampu mata kuliah tugas akhir penulis

yang selalu meluangkan waktu untuk memberi bimbingan, arahan, motivasi dan bantuan yang berharga kepada penulis agar selalu tetap semangat dalam menyelesaikan tugas akhir .

10. Bapak Dr. Abdurahman Dwijotomo, S.ST, M.Sc selaku dosen penguji II penulis yang memberikan pertanyaan kritis, masukan dan saran dalam penulisan tugas akhir ini.
11. Seluruh staff pengajar, laboran dan administrasi di jurusan Teknik Elektro Politeknik Negeri Batam.
12. Seluruh rekan - rekan Teknik Mekatronika angkatan 2020 yang telah membantu dan mendukung penulis.

Batam, 02 Januari 2024

Penulis

Daftar Isi

Pernyataan Keaslian Tugas Akhir	1
Lembar Pengesahan	2
Abstrak	3
<i>Abstract</i>	4
Kata Pengantar	5
Daftar Isi	7
Daftar Gambar	10
Daftar Tabel	13
Bab 1. Pendahuluan	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Tujuan	2
1.4. Manfaat	3
1.5. Batasan	3
1.6. <i>Work Breakdown Structure</i>	3
Bab 2. Tinjauan Pustaka	5
2.1. Sistem Monitoring	5
2.2. <i>Internet of Things</i>	5
2.2.1 <i>Ubidots</i>	6
2.2.2 <i>ThingSpeak</i>	7
2.3 <i>Quality of Service</i>	7
2.4 <i>Wireshark</i>	8
2.5 MQTT	8
2.6 RNN-LSTM (<i>Recurrent Neural Network - Long Short Term Memory</i>)	8
2.7 <i>Sensor Water Flow</i>	9
2.8 NodeMCU ESP32	10
2.9 <i>Software Arduino</i>	11
2.10 Relay	12
2.11 <i>Solenoid Valve</i>	12
Bab 3. Metodologi Penelitian	13

3.1 Perancangan Sistem	14
3.1.1 Perancangan Mekanikal	15
3.1.2 Perancangan Elektrikal	15
3.1.3 Perancangan <i>Software</i>	17
3.1.3.1 Platform IoT <i>Ubidots</i>	17
3.1.3.2 Platform IoT <i>ThingSpeak</i>	19
3.1.3.3 Prediksi Data Menggunakan RNN-LSTM	20
3.1.3.4 <i>Quality of Service</i> pada <i>Wireshark</i>	22
3.2 Pengujian	23
3.2.1 Pengujian Alat	23
3.2.2 Perbandingan Data Aktual dan Prediksi Data Menggunakan RNN-LSTM	24
3.2.3 Perbandingan Performa QoS (<i>Quality of Service</i>) pada Platform IoT <i>Ubidots</i> dan <i>ThingSpeak</i>	24
3.2.3.1 <i>Throughput</i>	25
3.2.3.2 <i>Jitter</i>	25
3.2.3.3 <i>Delay</i>	26
3.2.3.4 <i>Packet Loss</i>	27
Bab 4. Hasil dan Pembahasan	29
4.1. Pengujian Alat	29
4.1.1 Pengujian <i>Volume</i>	29
4.1.1.1 Pengujian 1	29
4.1.1.2 Pengujian 2	30
4.1.1.3 Pengujian 3, Pengujian 4, Pengujian 5 dan Pengujian 6	30
4.1.1.4 Pengujian 7	31
4.1.1.5 Pengujian 8	32
4.1.2 Pengujian <i>Flow Rate</i>	32
4.1.2.1 Pengujian 1	32
4.1.2.2 Pengujian 2	33
4.1.2.3 Pengujian 3, Pengujian 4, Pengujian 5, Pengujian 6, dan Pengujian 7	33
4.1.2.4 Pengujian 8	35
4.1.3 Error dan Akurasi	35

4.2. Perbandingan Data Aktual dan Prediksi Data Menggunakan RNN-LSTM	36
4.3 Perbandingan Performa QoS (<i>Quality of Service</i>) pada <i>Platform</i> IoT <i>Ubidots</i> dan <i>ThingSpeak</i>	38
4.3.1 Hasil Pembacaan <i>Throughput</i>	38
4.3.2 Hasil Pembacaan <i>Jitter</i>	39
4.3.3 Hasil Pembacaan <i>Delay</i>	40
4.3.4 Hasil Pembacaan <i>Packet Loss</i>	41
4.4. Pembahasan	42
Bab 5. Kesimpulan dan Saran	46
5.1. Kesimpulan	46
5.2. Saran	46
Daftar Pustaka	48
Lampiran	53
A. Aplikasi	53
B. Meteran Manual	58
C. Statistik QoS	62

Daftar Gambar

Gambar 1. Ilustrasi IoT[12]	6
Gambar 2. <i>Platform</i> IoT <i>Ubidots</i> [14]	6
Gambar 3. <i>Platform</i> IoT <i>ThingSpeak</i> [16]	7
Gambar 4. <i>Wireshark</i> [22]	8
Gambar 5. MQTT [24]	8
Gambar 6. Sensor <i>Water Flow</i> [30]	10
Gambar 7. <i>Mikrokontroler</i> NodeMCU ESP32 [31]	11
Gambar 8. <i>Pinout</i> NodeMCU ESP32 [31]	11
Gambar 9. Tampilan Arduino IDE	11
Gambar 10. Relay [34]	12
Gambar 11. <i>Solenoid Valve</i>	12
Gambar 12. <i>Flowchart</i> Pelaksanaan Program	13
Gambar 13. Blok Diagram Sistem	14
Gambar 14. Desain Mekanikal	15
Gambar 15. Desain Elektrikal	16
Gambar 16. <i>Flowchart</i> Sistem <i>Software</i>	17
Gambar 17. <i>Flowchart</i> Penggunaan <i>Platform</i> IoT <i>Ubidots</i>	19
Gambar 18. <i>Flowchart</i> Penggunaan <i>Platform</i> IoT <i>ThingSpeak</i>	20
Gambar 19. <i>Flowchart</i> RNN-LSTM	21
Gambar 20. <i>Flowchart</i> Penggunaan <i>Wireshark</i>	22
Gambar 21. Tampilan Rumus pada <i>Google Colaboratory</i>	24
Gambar 22. Pengujian 1 Grafik <i>Volume</i>	29
Gambar 23. Pengujian 2 Grafik <i>Volume</i>	30
Gambar 24. Pengujian 3 Grafik <i>Volume</i>	30
Gambar 25. Pengujian 4 Grafik <i>Volume</i>	30
Gambar 26. Pengujian 5 Grafik <i>Volume</i>	31
Gambar 27. Pengujian 6 Grafik <i>Volume</i>	31
Gambar 28. Pengujian 7 Grafik <i>Volume</i>	31
Gambar 29. Pengujian 8 Grafik <i>Volume</i>	32
Gambar 30. Pengujian 1 Grafik <i>Flow Rate</i>	32
Gambar 31. Pengujian 2 Grafik <i>Flow Rate</i>	33

Gambar 32. Pengujian 3 Grafik <i>Flow Rate</i>	33
Gambar 33. Pengujian 4 Grafik <i>Flow Rate</i>	34
Gambar 34. Pengujian 5 Grafik <i>Flow Rate</i>	34
Gambar 35. Pengujian 6 Grafik <i>Flow Rate</i>	34
Gambar 36. Pengujian 7 Grafik <i>Flow Rate</i>	34
Gambar 37. Pengujian 8 Grafik <i>Flow Rate</i>	35
Gambar 38. Hasil Perhitungan Matrik	36
Gambar 39. Grafik Data Prediksi dan Data Aktual	37
Gambar 40. Perhitungan pada <i>Throughput</i>	38
Gambar 41. Grafik Perbandingan <i>Throughput</i>	39
Gambar 42. Perhitungan pada <i>Jitter</i>	39
Gambar 43. Grafik Perbandingan <i>Jitter</i>	40
Gambar 44. Perhitungan pada <i>Delay</i>	40
Gambar 45. Grafik Perbandingan <i>Delay</i>	41
Gambar 46. Grafik Perbandingan <i>Packet Loss</i>	42
Gambar 48. <i>Ubidots</i> Pengujian 1	53
Gambar 49. <i>ThingSpeak</i> Pengujian 1	53
Gambar 50. <i>Ubidots</i> Pengujian 2	54
Gambar 51. <i>ThingSpeak</i> Pengujian 2	54
Gambar 52. <i>Ubidots</i> Pengujian 3	54
Gambar 53. <i>ThingSpeak</i> Pengujian 3	54
Gambar 54. <i>Ubidots</i> Pengujian 4	55
Gambar 55. <i>ThingSpeak</i> Pengujian 4	55
Gambar 56. <i>Ubidots</i> Pengujian 5	55
Gambar 57. <i>ThingSpeak</i> Pengujian 5	55
Gambar 58. <i>Ubidots</i> Pengujian 6	56
Gambar 59. <i>ThingSpeak</i> Pengujian 6	56
Gambar 60. <i>Ubidots</i> Pengujian 7	56
Gambar 61. <i>ThingSpeak</i> Pengujian 7	56
Gambar 62. <i>Ubidots</i> Pengujian 8	57
Gambar 63. <i>ThingSpeak</i> Pengujian 9	57
Gambar 64. Meteran Manual Sebelum Pengujian 1	58
Gambar 65. Meteran Manual Sesudah Pengujian 1	58

Gambar 66. Meteran Manual Sebelum Pengujian 2	58
Gambar 67. Meteran Manual Sesudah Pengujian 2	58
Gambar 68. Meteran Manual Sebelum Pengujian 3	59
Gambar 69. Meteran Manual Sesudah Pengujian 3	59
Gambar 70. Meteran Manual Sebelum Pengujian 4	59
Gambar 71. Meteran Manual Sesudah Pengujian 4	59
Gambar 72. Meteran Manual Sebelum Pengujian 5	60
Gambar 73. Meteran Manual Sesudah Pengujian 5	60
Gambar 74. Meteran Manual Sebelum Pengujian 6	60
Gambar 75. Meteran Manual Sesudah Pengujian 6	60
Gambar 76. Meteran Manual Sebelum Pengujian 7	61
Gambar 77. Meteran Manual Sesudah Pengujian 7	61
Gambar 78. Meteran Manual Sebelum Pengujian 8	61
Gambar 79. Meteran Manual Sesudah Pengujian 8	61
Gambar 80. <i>Ubidots</i> Pengujian 1	62
Gambar 81. <i>ThingSpeak</i> Pengujian 1	62
Gambar 82. <i>Ubidots</i> Pengujian 2	62
Gambar 83. <i>ThingSpeak</i> Pengujian 2	62
Gambar 84. <i>Ubidots</i> Pengujian 3	63
Gambar 85. <i>ThingSpeak</i> Pengujian 3	63
Gambar 86. <i>Ubidots</i> Pengujian 4	63
Gambar 87. <i>ThingSpeak</i> Pengujian 4	63
Gambar 88. <i>Ubidots</i> Pengujian 5	64
Gambar 89. <i>ThingSpeak</i> Pengujian 5	64
Gambar 90. <i>Ubidots</i> Pengujian 6	64
Gambar 91. <i>ThingSpeak</i> Pengujian 6	64
Gambar 92. <i>Ubidots</i> Pengujian 7	65
Gambar 93. <i>ThingSpeak</i> Pengujian 7	65
Gambar 94. <i>Ubidots</i> Pengujian 8	65
Gambar 95. <i>ThingSpeak</i> Pengujian 8	65

Daftar Tabel

Tabel 1. <i>Work Breakdown Structure</i>	4
Tabel 2. Konfigurasi <i>Wiring</i>	16
Tabel 3. Pengujian Sensor <i>Water Flow</i>	23
Tabel 4. Kategori <i>Throughput</i> [36]	25
Tabel 5. Kategori <i>Jitter</i> [36]	26
Tabel 6. Kategori <i>Delay</i> [36]	26
Tabel 7. Kategori <i>Packet Loss</i> [36]	27
Tabel 8. Metode Tabel Perbandingan Performa QoS pada <i>Platform IoT Ubidots</i> dan <i>ThingSpeak</i>	27
Tabel 9. Pengujian Sensor <i>Water Flow</i>	35
Tabel 10. Standar Persentase dan Nilai QoS [31]	42
Tabel 11. Rata-Rata Parameter QoS pada <i>Ubidots</i> dan <i>ThingSpeak</i>	43
Tabel 12. Parameter pada RNN-LSTM	44

Bab 1. Pendahuluan

1.1. Latar Belakang

Seiring dengan pesatnya kemajuan teknologi, sistem pengukuran terus mengalami kemajuan yang signifikan. Hal ini telah menghasilkan variasi metode pengukuran debit air dengan metode digital. Meskipun demikian, masih banyak orang yang mengukur debit air menggunakan meteran manual yang biasanya digunakan pada perumahan [1].

LSTM merupakan perkembangan arsitektur RNN yang dirancang khusus untuk menangani data berurutan, dimana masukan berupa urutan data atau *sequence of items* dan menghasilkan vektor yang merangkum urutan tersebut [2]. Model ini memiliki kemampuan untuk mengingat informasi dari urutan sebelumnya. RNN-LSTM digunakan sebagai alat untuk menganalisis prediksi data aktual (manual) dan data aplikasi dengan menggunakan sensor YF-S201.

Diera yang ditunjukkan oleh kemajuan teknologi yang sangat pesat, penggunaan IoT (*Internet of Things*) telah mempengaruhi berbagai aspek kehidupan [3]. Konsep IoT diterapkan pada objek-objek yang dilengkapi dengan teknologi sensor dan perangkat lunak guna memungkinkan interaksi, pengontrolan, konektivitas, dan pertukaran data dengan perangkat lain melalui internet [4]. *Ubidots* dan *ThingSpeak* merupakan contoh teknologi IoT yang saat ini tersedia.

Ubidots adalah sebuah *platform* yang memiliki kemampuan untuk mengimplementasikan prinsip IoT dengan tujuan utama untuk mengumpulkan data dari berbagai sumber *input* dan mengatur tindakan yang diinginkan pada *output* yang ditetapkan [5]. Sementara itu, *ThingSpeak* merupakan sebuah *platform open source* yang berbentuk *website*. *Platform* ini menyediakan layanan IoT dan mampu menerima data melalui protokol HTTP melalui koneksi internet [6]. Namun, meskipun kedua *platform* ini sangat populer, belum banyak penelitian secara lengkap untuk menganalisis performansi pengiriman data menggunakan pendekatan QoS.

Quality of Service (QoS) adalah kemampuan jaringan untuk memberikan layanan yang optimal dengan mempertimbangkan aspek *bandwidth*, *jitter*, dan *delay*. Dalam jaringan IP, QoS berperan penting dalam menyajikan kualitas layanan yang beda berdasarkan kebutuhan yang beragam [7]. Perusahaan penyedia layanan internet, atau *Internet Service Provider* (ISP) bersaing untuk menyediakan layanan yang optimal kepada pelanggan mereka untuk memenuhi kebutuhan tersebut [8].

Berdasarkan hasil penelitian sebelumnya dan jurnal yang paling sering ditemukan, banyak yang memanfaatkan 3 parameter sebagai *metrics* pengukuran QoS, yaitu *bandwidth*, *jitter* dan *delay*. Oleh karena itu, dalam penelitian ini kami

memperluas cakupan dengan menggunakan 4 parameter yaitu *throughput*, *jitter*, *delay* dan *packet loss*. Penelitian ini bertujuan membandingkan *platform* IoT *Ubidots* dan *platform* IoT *ThingSpeak* berdasarkan 4 parameter tersebut.

Dalam penelitian ini, kami menggabungkan beberapa pendekatan dari jurnal sebelumnya. Kami menggunakan *platform* IoT *Ubidots* dan *ThingSpeak* dengan alat sederhana yang diberi nama *Mow Flow* terdiri dari sensor *water flow* YF-S201 untuk mengukur *volume* dan *flow rate* yang dikontrol menggunakan relay untuk membuka *solenoid valve* serta ESP 32 sebagai mikrokontrolernya. Penelitian ini berfokus pada pengukuran QoS *metrics* kedua *platform* IoT untuk memastikan pengiriman data sesuai dengan yang dikirimkan dan memungkinkan memprediksi *volume* menggunakan data aktual dan data aplikasi yang lebih akurat untuk analisis yang efektif dengan penerapan RNN-LSTM berdasarkan data sensor YF-S201. Sehingga dari uraian diatas ini maka penulis mengambil judul Tugas Akhir “Analisis Performansi Pengiriman Data Menggunakan Pendekatan QoS *Metrics* pada *Platform* IoT *Ubidots* dan *ThingSpeak*, Studi Kasus: *Monitoring* Aliran Air Rumah Pintar”.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, maka dapat dirumuskan permasalahan sebagai berikut.

1. Bagaimana merancang dan membuat sistem *monitoring* aliran air pada rumah pintar menggunakan sensor *water flow*?
2. Bagaimana performansi pengiriman data pada *platform* IoT *Ubidots* dan *ThingSpeak* dalam konteks *monitoring* aliran air rumah pintar?
3. Bagaimana penerapan RNN-LSTM untuk memprediksi data dari data aktual (data manual) dan data sensor YF-S201 pada *platform* IoT *Ubidots* dan *ThingSpeak*?
4. Bagaimana pengaruh penggunaan pendekatan QoS *metrics* terhadap performansi pengiriman data pada *platform* IoT *Ubidots* dan *ThingSpeak* dalam konteks aliran air rumah pintar?
5. Bagaimana hasil pengujian dari alat *monitoring* aliran air rumah pintar dan hasil dari perbandingan performansi dalam pengiriman data dari *platform* IoT *Ubidots* dan *ThingSpeak*?

1.3. Tujuan

Adapun tujuan dari proposal tugas akhir ini adalah :

1. Merancang sistem dan alat *monitoring* aliran air pada rumah pintar.
2. Menganalisis performansi pengiriman data sensor pada *platform* IoT *Ubidots* dan *ThingSpeak* dalam konteks aliran air rumah pintar.

3. Menganalisis penerapan RNN-LSTM dalam memprediksi data sensor pada data aktual (data manual) dan data aplikasi di masa akan datang
4. Mengidentifikasi penggunaan pendekatan QoS *metrics* terhadap performansi pengiriman data pada *platform* IoT *Ubidots* dan *ThingSpeak* dalam konteks aliran air rumah pintar.
5. Mengamati dan menganalisis hasil dari pengujian alat *monitoring* aliran air pada rumah pintar dan hasil dari perbandingan performansi dalam pengiriman data dari *platform* IoT *Ubidots* dan *ThingSpeak*.

1.4. Manfaat

Manfaat dari hadirnya Analisis Performansi Pengiriman Data Menggunakan Pendekatan QoS *Metrics* pada *Platform* IoT *Ubidots* dan *ThingSpeak*, Studi Kasus: *Monitoring* Aliran Air Rumah Pintar adalah menciptakan sebuah alat yang dapat mengukur debit aliran air yang digunakan untuk mengirimkan data secara *real time* ke *platform* IoT *Ubidots* dan *ThingSpeak* dengan kondisi yang sama dan membandingkan performa *website Ubidots* dan *ThingSpeak* agar dapat melihat keunggulan dan keterbatasan dari kedua platform sehingga dapat memutuskan *platform* mana yang lebih sesuai kebutuhan serta menganalisis data sensor YF-S201 dalam memprediksi volume pada data aktual (data manual) dan data aplikasi di masa akan datang menggunakan penerapan RNN-LSTM

1.5. Batasan

Berdasarkan latar belakang di atas agar tidak terjadi perluasan masalah, maka peneliti membatasi permasalahan penelitian ini sebagai berikut :

1. Perancangan sistem dan membuat alat *monitoring* aliran air pada rumah pintar menggunakan NodeMCU ESP32 dan sensor *water flow*.
2. Hasil pengujian dari alat *monitoring* aliran air dengan parameter *output* menggunakan tampilan *website*.
3. Perbandingan performansi dalam pengiriman data dan pengukuran QoS *metrics* dari *platform* IoT *Ubidots* dan *ThingSpeak*.
4. Analisis perbandingan pengiriman data pada *platform* IoT *Ubidots* dan *ThingSpeak* For Free.

1.6. Work Breakdown Structure

Penelitian ini pengerjaan alat dilakukan secara tim, dimana dalam satu tim terdapat dua anggota dengan pembagian tugas yang ada pada Tabel 1.

Tabel 1. Work Breakdown Structure

No.	Nama	Tugas dan Tanggung Jawab dalam Tim
1	Asri Arganing Hasanah	Pemrograman <i>Platform IoT Ubidots</i> , Pemrograman Mikrokontroler, Pengujian, Analisis QoS dan RNN-LSTM.
2	Nurul Khairani	Pemrograman <i>Platform IoT ThingSpeak</i> , Mekanikal, Elektrikal dan Analisis <i>Monitoring</i>

Bab 2. Tinjauan Pustaka

2.1. Sistem *Monitoring*

Sistem adalah kumpulan komponen yang bekerja sama untuk mencapai suatu tujuan tertentu. Sistem juga dapat didefinisikan sebagai kumpulan bagian yang unsur-unsurnya saling terkait satu sama lain. Dalam pengertian ini, sistem adalah kumpulan objek yang berisi hubungan fungsional antara setiap fitur dan hubungan dan komponen yang saling berinteraksi satu sama lain untuk mencapai tujuan tertentu.

Monitoring adalah kegiatan yang memastikan terpenuhinya semua tujuan yang ditetapkan oleh organisasi dan manajemen. *Monitoring* dapat ditafsirkan sebagai suatu langkah untuk menilai apakah kegiatan dilaksanakan sesuai dengan rencana atau tidak, mengidentifikasi masalah yang muncul untuk segera diperbaiki dan mengevaluasi apakah pengelolaan yang digunakan sudah tepat atau belum.

Sesuai definisi di atas, sistem *monitoring* adalah pemeriksaan dan evaluasi data tentang kinerja proyek atau kegiatan dengan menilai apakah terjadi peningkatan kepatuhan peraturan atau tidak [9]. *Monitoring* ada 2 macam, yaitu [10]:

1. *Monitoring* langsung adalah pengawasan yang dilaksanakan melalui kunjungan dan pemeriksaan objek yang dipantau secara langsung.
2. *Monitoring* tidak langsung adalah pengawasan yang dilakukan tanpa memerlukan objek yang dipantau secara jarak jauh.

2.2. *Internet of Things*

Istilah *Internet of Things* (IoT) adalah sebuah konsep yang dicetuskan Kevin Ashton di tahun 1999. IoT merujuk pada kumpulan objek yang saling terhubung dengan internet. *Things* dapat berbentuk *tags*, sensor, manusia, aktuator dan lainnya. Peran IoT adalah untuk menyatukan data dan informasi tentang lingkungan kemudian memproses data itu sehingga data tersebut akurat. IoT merupakan konsep yang bermaksud untuk mengembangkan manfaat dari konektivitas internet yang selalu aktif. Fungsinya antara lain berbagi data, mengendalikan dari kejauhan, dan seterusnya, meliputi objek nyata [11]. Ilustrasi IoT dapat dilihat pada Gambar 1.

2.2.2 ThingSpeak

ThingSpeak merupakan *platform* IoT untuk menyimpan dan mengakses data dari beragam perangkat menggunakan protokol HTTP melalui Internet atau jaringan lokal. *ThingSpeak compatible* dengan berbagai perangkat seperti Arduino, Raspberry Pi, hingga ESP [15]. *Platform* ini menawarkan layanan secara gratis dan berbayar.



Gambar 3. Platform IoT ThingSpeak [16]

Platform ThingSpeak dirancang untuk mengakses data yang dikirim melalui mikrokontroler secara *real-time*. Pada *ThingSpeak*, langkah pertama yaitu membuat kanal untuk meluncurkan dan memodifikasi sesuai proyek yang dibuat. Dengan satu akun, pengguna dapat membuat banyak kanal sehingga sesuai untuk memuat banyak proyek [17]. Keuntungan dari *platform* ini adalah gratis, mudah untuk menampilkan data menggunakan diagram garis, dapat dihubungkan dengan *hardware* seperti mikrokontroler, dan relatif mudah untuk digunakan. Kekurangannya adalah ada beberapa fitur yang mengharuskan pengguna meningkatkan akun pengguna untuk mendapatkan batas data saluran API dan membuat grafik [18].

2.3 Quality of Service

Quality of Service (QoS) ukuran kemampuan jaringan dalam menyediakan layanan untuk lalu lintas data tertentu. Salah satu contoh jaringan yang dimaksud adalah jaringan internet protokol (IP). Analisis QoS sangat penting dalam sistem sistem jaringan internet karena [19]:

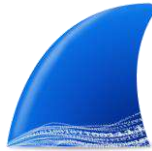
- a. Memberikan prioritas pada aplikasi penting dalam jaringan.
- b. Memaksimalkan pemanfaatan investasi infrastruktur jaringan yang tersedia.
- c. Meningkatkan hasil investasi infrastruktur jaringan yang dikerjakan.
- d. Meningkatkan kinerja pada aplikasi yang sensitif terhadap latensi seperti audio dan video.
- e. Menanggapi perubahan trafik internet.

Parameter yang menentukan QoS antara lain *throughput*, *packet loss*, *delay*, dan *jitter*. Standar untuk mengukur kinerja jaringan adalah TIPHON (*Telecommunications and Internet Protocol Harmonization Over Network*).

TIPHON adalah standar yang dipublikasikan oleh *European Telecommunications Standards Institute* (ETSI) untuk mengevaluasi parameter QoS [20].

2.4 Wireshark

Wireshark adalah perangkat lunak yang digunakan untuk menganalisa aktivitas jaringan komputer. *Wireshark* memberikan manfaat yang signifikan bagi individu yang bekerja dalam bidang jaringan, termasuk profesional jaringan, administrator, peneliti dan pengembang perangkat lunak jaringan. *Wireshark* memiliki kapabilitas *real-time* dalam merekam paket-paket informasi yang bergerak di jaringan. Perangkat ini mampu menangkap dan menganalisis berbagai jenis paket informasi yang dikirim dalam berbagai protokol dengan gampang [21].



Gambar 4. *Wireshark* [22]

2.5 MQTT

Protokol MQTT merupakan protokol yang berbasis TCP/IP *stack* dan memiliki ukuran paket data dengan *overhead* yang kecil (minimum 2 bytes), sehingga dampaknya pada konsumsi listrik yang juga kecil [23].



Gambar 5. MQTT [24]

Protokol ini bertipe *data-agnostic*, yang berarti pengguna dapat mengirimkan data apapun, seperti biner, text dan XML atau JSON, protokol ini juga menggunakan model *publish/subscribe* daripada model client-server [23].

2.6 RNN-LSTM (*Recurrent Neural Network - Long Short Term Memory*)

RNN-LSTM merupakan kombinasi unit *Recurrent Neural Network* (RNN) dan *Long Short-Term Memory* (LSTM). Dengan menggunakan LSTM, RNN-LSTM dapat lebih efisien dalam memproses data deret waktu dengan mengatasi beberapa kendala yang dihadapi RNN tradisional. Hal ini menjadikannya pilihan populer untuk tugas-tugas seperti prediksi deret waktu, terjemahan bahasa, dan tugas-tugas lain yang melibatkan data sekuensial [25].

RNN-LSTM digunakan untuk menganalisis yang efektif dan membuat prediksi besar volume pada data aktual dan data aplikasi berdasarkan dari data sensor aliran YF-S201. Dengan menggunakan model ini, kita dapat menggali pola yang mungkin sulit dibaca secara manual.

Mean Square Error (MSE) merupakan metrik yang mengukur seberapa besar perbedaan antara nilai prediksi oleh model dengan nilai aktual dalam suatu *dataset*. Makin rendah angka MSE, maka semakin baik model tersebut. Berikut ini rumus untuk menghitung MSE [26].

$$MSE = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n} \quad (1)$$

Mean Absolute Error (MAE) adalah matrik yang mengukur rata-rata kesalahan absolute antara nilai yang diprediksi oleh model dan nilai aktual dalam suatu *dataset*. Semakin rendah nilai MAE, semakin bagus kinerja modelnya. Rumus berikut menghitung MAE [27].

$$MAE = \frac{\sum_{i=1}^n |Y_i - \hat{Y}_i|}{n} \quad (2)$$

Koefisien determinasi, juga dikenal sebagai koefisien *R-square* merupakan sejauh mana model regresi menjelaskan variasi variabel dependen. *R-square* adalah skor bebas skala, artinya nilai *R-square* selalu berada dalam rentang 0 hingga 1, terlepas dari besarnya variabel-variabel yang terlibat. Semakin mendekati 1, semakin besar kontribusi model terhadap menjelaskan variabel dependen [28].

$$R^2 = 1 - \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad [29] \quad (3)$$

Keterangan :

Y_i = Data sebenarnya

\hat{Y}_i = Nilai Prediksi dari Variabel Y

n = Banyaknya Observasi

2.7 Sensor Water Flow

Sensor *water flow* merupakan sensor yang menghitung aliran air dan menjalankan mesin dalam satuan liter. Motor berputar sesuai laju air yang mengalir. Sensor *water flow* ini terdiri dari katup plastik, rotor air, dan sensor efek Hall. Prinsip kerjanya memanfaatkan efek Hall [1].



Gambar 6. Sensor Water Flow [30]

Sensor efek *Hall* membaca sinyal tegangan dalam bentuk pulsa kemudian mengirimkan sinyal ke mikrokontroler. Rotor magnet akan berputar pada kecepatan tertentu sesuai dengan flow rate dikarenakan air yang mengalir melewati katup plastik sensor *water flow*. Keluaran dari pulsa tegangan berada pada level yang sama dengan *input* yang dimiliki frekuensi flow rate air. Sinyal kemudian diolah menjadi data digital oleh mikrokontroler. Kecepatan air yang berasal dari *water flow* sensor ditentukan dengan rumus berikut [1]:

$$Q = \frac{\text{Jumlah pulse/menit}}{7,5} \quad (4)$$

Nilai konstanta 7,5 adalah konstanta frekuensi dalam *datasheet* sensor, sementara untuk mencari volume air dalam meter kubik, gunakan rumus berikut:

$$V = \frac{Q}{\frac{60}{1000}} \quad (5)$$

Keterangan :

V = Jumlah volume air yang terukur m³

Q = Debit air (m³/s)

60 = Detik (s)

1000 = Nilai Liter dibagi m³

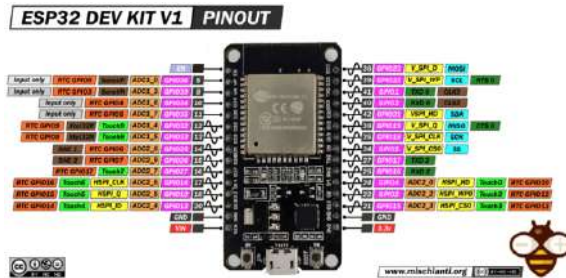
2.8 NodeMCU ESP32

ESP32 merupakan mikrokontroler penerus ESP8266 memiliki *WiFi* dan *Bluetooth* sebagai keunggulan yang memudahkan dalam pembuatan sistem IoT yang membutuhkan koneksi *wireless* fitur-fitur tersebut tidak dapat ditemukan dalam ESP8266 maka dari itu ESP32 disebut sebagai *upgrade* dari ESP8266 [31].



Gambar 7. Mikrokontroler NodeMCU ESP32 [31]

Berikut ini adalah pinout ESP32 Dev Kit V1:



Gambar 8. Pinout NodeMCU ESP32 [31]

2.9 Software Arduino

Arduino IDE adalah perangkat lunak yang dimanfaatkan untuk pemrograman pada mikrokontroler. Arduino IDE terdiri dari bahasa pemrograman JAVA yang dilengkapi dengan *library C/C++ (wiring)*, yang menyederhanakan operasi *input/output* [32].



Gambar 9. Tampilan Arduino IDE

2.10 Relay

Relay atau saklar bekerja secara otomatis menggunakan aliran listrik dan merupakan komponen elektromekanik yang terdiri dari dua bagian utama yaitu, *Coil* dan seperangkat kotak saklar. Prinsip kerja relay yaitu elektromagnetik menggerakkan kontak saklar dengan arus listrik yang rendah dan mengalirkan ke listrik bertegangan besar [33].



Gambar 10. Relay [34]

2.11 Solenoid Valve

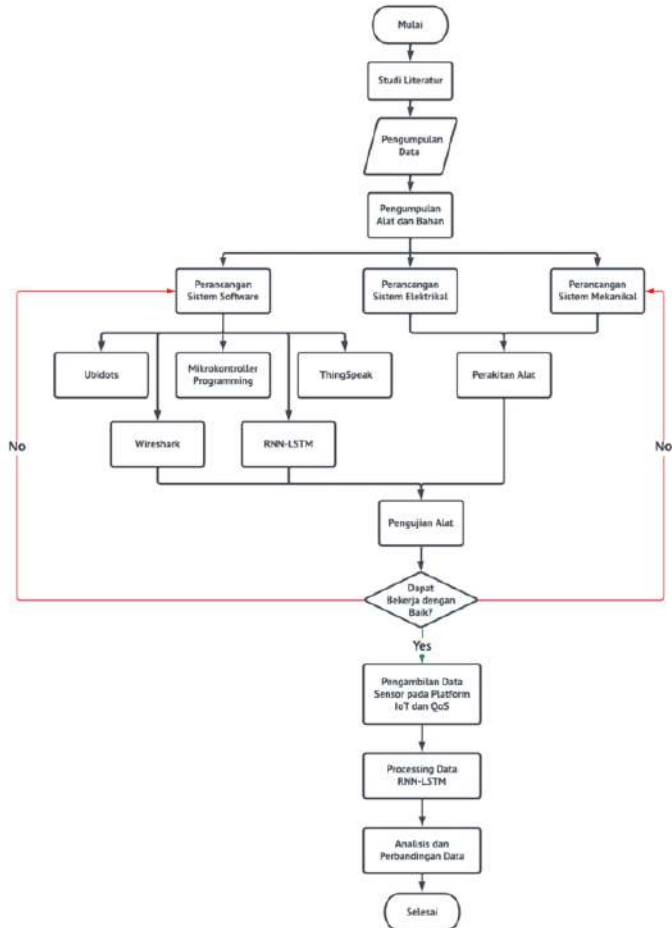
Solenoid Valve merupakan katup yang dikendalikan oleh arus listrik. *Solenoid Valve* merupakan kontrol yang paling sering digunakan dalam sistem fluida seperti sistem pneumatik dan sistem hidrolik. *Solenoid valve* berfungsi untuk menyalurkan udara bertekanan menuju aktuator dan tangki air yang membutuhkan *solenoid valve* untuk mengatur pengisian air [35].



Gambar 11. *Solenoid Valve*

Bab 3. Metodologi Penelitian

Pelaksanaan penelitian “Analisis Performansi Pengiriman Data Menggunakan Pendekatan QoS Metrics pada Platform IoT Ubidots dan ThingSpeak, Studi Kasus: Monitoring Aliran Air Rumah Pintar” terdiri dari konfigurasi perangkat keras dan perancangan perangkat lunak dapat dilihat pada Gambar 12 yang menjelaskan secara urutan pelaksanaan kegiatan.

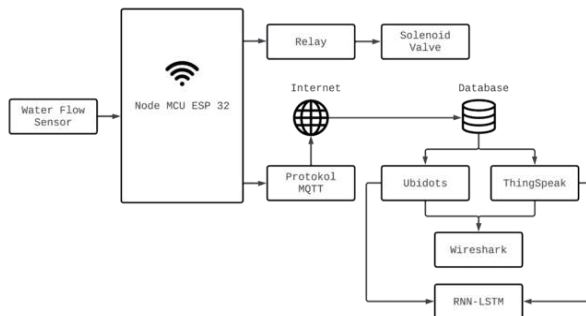


Gambar 12. Flowchart Pelaksanaan Program

Berdasarkan Gambar 12, proses dimulai dengan studi literatur dan pengumpulan data. Kemudian, tahap pembelian alat dan bahan. Selanjutnya tahap perancangan dengan 3 tahapan: *software*, elektrikal dan mekanikal. Pada perancangan *software*, penulis membuat program untuk menjalankan sensor dan mikrokontroler serta mengirimkan data ke platform *Ubidots* dan *ThingSpeak*. Menginstall aplikasi *wireshark* dan membuat program untuk proses data pada RNN-LSTM. Tahap elektrikal dan mekanikal mencakup perakitan alat. Jika hasil pengujian sesuai dengan standar dan berjalan dengan baik, maka proses selanjutnya adalah pengambilan data. Proses ini mencakup pemantauan pengiriman data pada kedua *platform* IoT dan pemantauan menggunakan *wireshark* untuk memeriksa QoS yang dihasilkan kedua *platform*. Data yang diperoleh dari sensor pada kedua *platform* IoT akan langsung diproses menggunakan model RNN-LSTM. Setelah itu, dilakukan analisis dan perbandingan data hasil pengujian alat, QoS dan prediksi RNN-LSTM. Namun, jika hasil pengujian tidak sesuai dengan prosedur yang ditetapkan, maka pekerjaan akan kembali ke tahap pemeriksaan perancangan *software*, elektrikal dan mekanikal untuk memperbaiki sebelum melanjutkan tahapan selanjutnya.

3.1 Perancangan Sistem

Pada rangkaian sistem “Analisis Performansi Pengiriman Data Menggunakan Pendekatan QoS Metrics pada Platform IoT Ubidots dan ThingSpeak, Studi Kasus: Monitoring Aliran Air Rumah Pintar” terdiri dari satu set alat yang sudah kami rangkai untuk mengirimkan data ke dua *platform* IoT, yaitu *Ubidots* dan *ThingSpeak*. Blok diagram sistem untuk rangkaian dapat dilihat pada gambar dibawah ini



Gambar 13. Blok Diagram Sistem

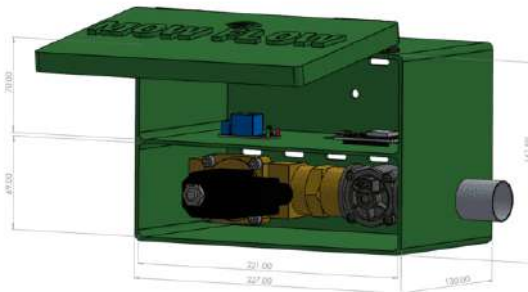
Pada Gambar 13 menjelaskan bahwa sistem ini menggunakan NodeMCU ESP32 sebagai mikrokontroler dengan pin GPIO 15 digunakan untuk mengirimkan data dari sensor *water flow* YF-S201, sedangkan pin GPIO 13 digunakan untuk mengirimkan data dari relay. Sistem ini mempunyai 1 *inputan* yaitu sensor

waterflow sebagai penghitung volume dan *flow rate*. Kemudian NodeMCU ESP32 *menginput* data dan memproses hasilnya menuju output. Ada 3 *output* yaitu relay, *solenoid valve*, dan protokol MQTT. Pengiriman data volume dan *flow rate* melalui wifi ke NodeMCU ESP32, yang meminta alamat melalui protokol MQTT dengan mode internet untuk mengirimkan data debit air disimpan dalam *database* kemudian ditampilkan pada aplikasi *Ubidots* dan *ThingSpeak*. *Ubidots* dan *ThingSpeak* akan menampilkan data melalui *website* apabila ada layanan internet dimana informasi yang ditampilkan berupa grafik data volume dan *flow rate* air dari pembacaan sensor *waterflow* YF-S201. Selama proses pengiriman data ke *platform* IoT, *Wireshark* akan terus memantau data yang dikirimkan untuk mengukur QoS.

Setelah data pengiriman berhasil diterima oleh masing-masing *platform* IoT, data akan diolah kembali dengan menggunakan model RNN-LSTM pada *google colab*, guna melakukan prediksi volume pada data aktual (data manual) dan data aplikasi yang didapat dari data sensor YF-S201 dalam beberapa periode waktu kedepan.

3.1.1 Perancangan Mekanikal

Dalam penelitian ini, perancangan mekanikal kami menggunakan *software Solidworks* 2018 untuk merancang desain awal dan untuk realisasinya kami menggunakan pipa paralon dan *filament* PLA+ sebagai box elektrikal nya. Dimensi dari box elektrikal ini adalah 22.7cm*13cm*14.75cm. Berikut ini merupakan desain dari *Mow Flow* pada Gambar 14.

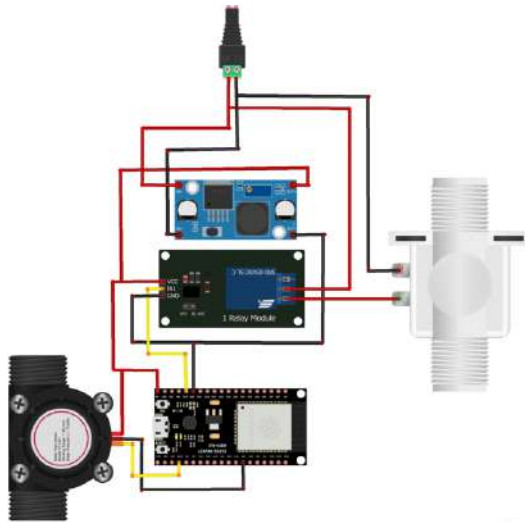


Gambar 14. Desain Mekanikal

3.1.2 Perancangan Elektrikal

Penelitian ini perancangan elektrikal dimulai dengan membuat rancangannya pada aplikasi *Fritzing*. Selanjutnya memastikan kondisi baik dari semua komponen dan dilanjutkan dengan pemasangan *wiring* secara tepat pada setiap komponen

untuk mencegah terjadinya gangguan. Desain elektrikal ditunjukkan pada Gambar 15.



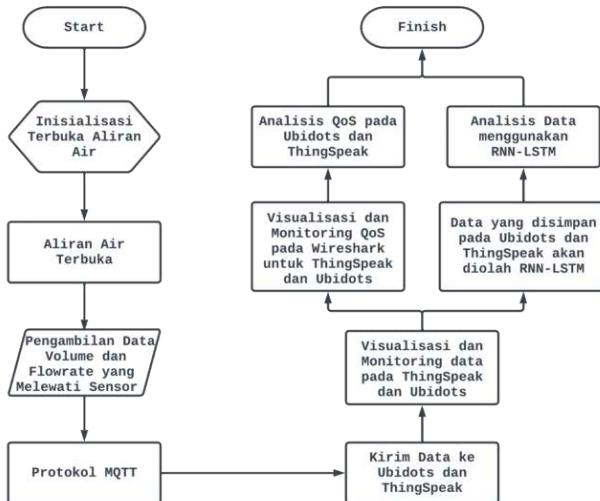
Gambar 15. Desain Elektrikal

Tabel 2. Konfigurasi *Wiring*

Komponen	Pin Sensor	ESP 32	Lainnya
Sensor Water Flow	VCC	5V	-
	GND	GND	-
	PIN	15	-
Module Relay	VCC	5V	-
	GND	GND	
	PIN	13	
Solenoid Valve	VCC	-	12 V (Adapter)
	PIN	-	NO Relay

Step Down	IN + / IN-	-	12 V (Adapter)
	OUT +	-	VCC Relay (5V)
	OUT -	-	GND (Relay)

3.1.3 Perancangan Software



Gambar 16. Flowchart Sistem Software

Penelitian ini ESP32 akan mengirimkan perintah ke relay untuk membuka *solenoid valve*, setelah itu air akan mengalir pada sensor *water flow* lalu data akan terkirim secara *realtime* ke *platform Ubidots* dan *ThingSpeak* melalui protokol MQTT. Data divisualisasikan dalam bentuk grafik dan numerik pada kedua *platform* sehingga kita dapat memantau nilai volume dan *flow rate* secara *realtime*. Selanjutnya ada 2 tahap berikutnya. Pertama, visualisasi dan *monitoring* QoS pada *Wireshark* untuk *Ubidots* dan *Thingspeak* dan dilanjutkan dengan menganalisis QoS kedua *platform* tersebut dan akan diamati *platform* mana yang akan mengirimkan data lebih baik cepat dan lebih baik sesuai dengan parameter QoS yang sudah ditentukan. Sedangkan tahap kedua, data volume yang disimpan pada *Ubidots* dan *ThingSpeak* akan diolah dan dibandingkan dengan data aktual

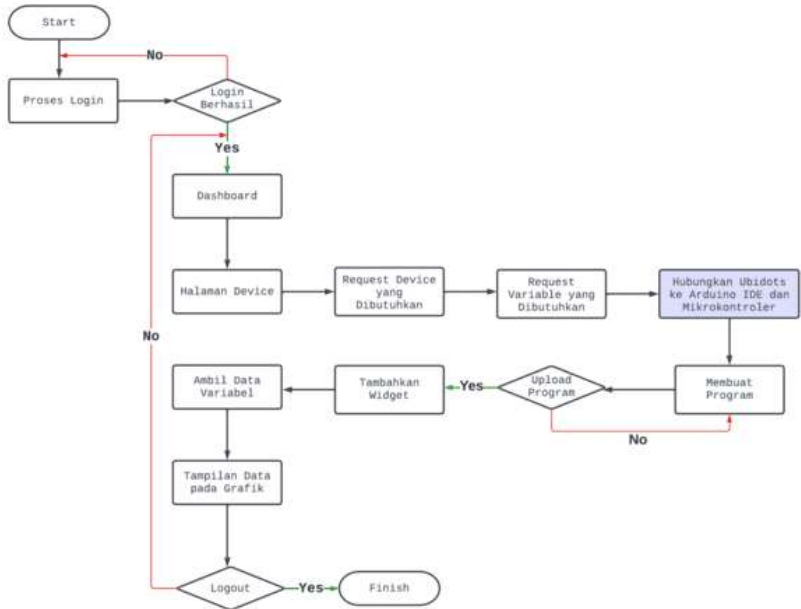
(data manual) dengan model RNN-LSTM untuk memprediksi pada data aktual dan data aplikasi dan dilanjutkan dengan menganalisis data RNN-LSTM, dan selesai.

3.1.3.1 Platform IoT Ubidots

Dalam penelitian ini, kami menggunakan *Ubidots Stem* yang memiliki kekurangan dan kelebihan. Berikut ini kekurangan dan kelebihan pada *Ubidots Stem* [40]. Layanan ini dapat diakses secara gratis.

- Kelebihan:
 - a. Dapat digunakan secara gratis untuk penggunaan non-komersial.
 - b. Mendukung hingga 3 perangkat.
 - c. Kapasitas penyerapan data hingga 4.000 titik per hari.
 - d. Retensi data hingga 1 bulan.
 - e. Tersedia dasbor statis dan dinamis.
 - f. Mendukung protokol HTTP, MQTT, TCP, dan UDP.
 - g. Memiliki *interval update* pesan terbatas minimum 1 *request/detik*.
- Kekurangan:
 - a. Hanya 1 *dashboard* dengan maksimal 10 *widget*.
 - b. Variabel terbatas hingga 10 per perangkat.
 - c. Plugin hanya 1 dengan 10.000 eksekusi per bulan.
 - d. Tidak ada perjanjian tingkat layanan.
 - e. Dukungan komunitas, tidak ada dukungan langsung.

Untuk mengakses *platform IoT Ubidots*, pengguna perlu *login* dengan menggunakan alamat *email* pribadi dan kata sandi yang telah kita daftarkan agar pengguna dapat mengakses *website*. *Dashboard Ubidots* berisi informasi *volume* dan *flowrate* aliran air yang ditampilkan dalam bentuk grafik dan angka.



Gambar 17. Flowchart Penggunaan Platform IoT Ubidots

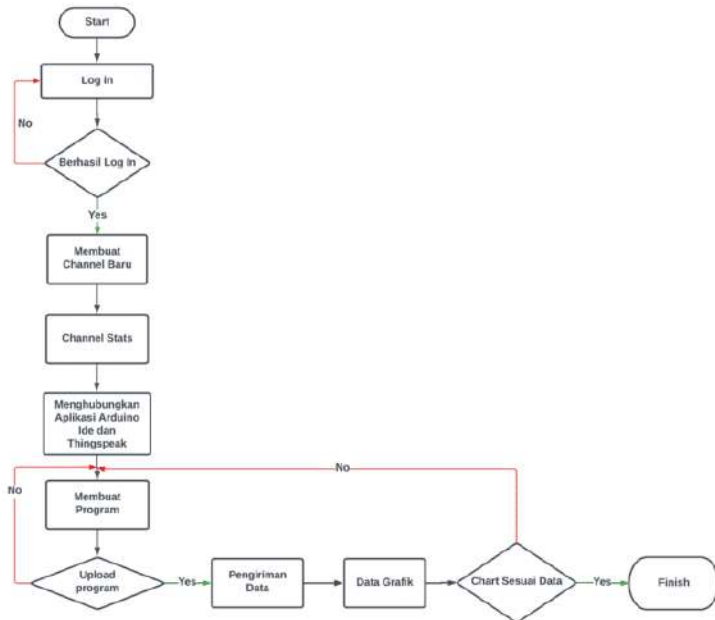
Gambar 17. Menunjukkan langkah untuk mengakses *website Ubidots*. Pertama *user* harus *login* dengan menggunakan alamat *email* dan kata sandi yang telah kita daftarkan agar pengguna dapat mengakses *website*. Setelah berhasil masuk ke menu utama, pilih opsi “*Devices*” dan klik “*Add New Device*” untuk membuat perangkat baru, lalu beri nama sesuai dengan keinginan *user*. Setelah itu, buat *variable* sesuai dengan keinginan *user*. Pilih *device* yang sudah dibuat sebelumnya sesuai dengan target *monitoring*. Pilih “*Add Devices*” untuk menambah *variabel* baru. Beri nama di setiap *variabel* yang dibuat disesuaikan dengan kebutuhan target *monitoring*. Selanjutnya hubungkan *website Ubidots* ke arduino IDE dan mikrokontroler dengan membuat program. *Upload* program yang sudah dibuat, jika gagal periksa kembali program yang sudah dibuat. Langkah berikutnya mengakses *Ubidots* untuk memeriksa data yang telah diterima dengan cara *refresh* pada bagian *dashboard*. Agar data terlihat dengan jelas, *user* perlu menambahkan *widget* sesuai dengan kebutuhan. Setelah itu, informasi yang sudah dikirim ke situs *web Ubidots* akan menampilkan target *monitoring* yaitu informasi *volume* dan *flowrate* aliran air yang ditampilkan dalam bentuk grafik dan angka. Jika program tidak bekerja dengan baik, maka proses akan kembali ke proses awal.

3.1.3.2 Platform IoT ThingSpeak

Dalam penelitian ini, kami menggunakan *ThingSpeak For Free* yang memiliki kekurangan dan kelebihan. Berikut ini kekurangan dan kelebihan pada *ThingSpeak For Free* [41]:

- Kelebihan :
 - a. Dapat digunakan secara gratis untuk penggunaan non-komersial.
 - b. Dapat menyimpan hingga 10 juta pesan.
 - c. Terintegrasi dengan MATLAB untuk analisis dan visualisasi data.
- Kekurangan:
 - a. Hanya terbatas untuk penggunaan non-komersial.
 - b. Hanya bisa membuat maksimal 4 *channel*.
 - c. *Interval update* pesan terbatas minimum 15 detik.
 - d. Tidak mendapatkan dukungan teknis.
 - e. *Timeout* komputasi MATLAB lebih singkat.

Untuk menggunakan *website ThingSpeak* kita harus membuat akun untuk mendapatkan akses *website*. Dalam mendaftarkan akun perlu menggunakan *email* pribadi, setelahnya penulis dapat mengikuti pengarahan dari *ThingSpeak*. Pada bagian ini penulis diminta untuk memasukkan email, *verifikasi* email, dan pembuatan kata sandi untuk bisa mengakses *website ThingSpeak*.

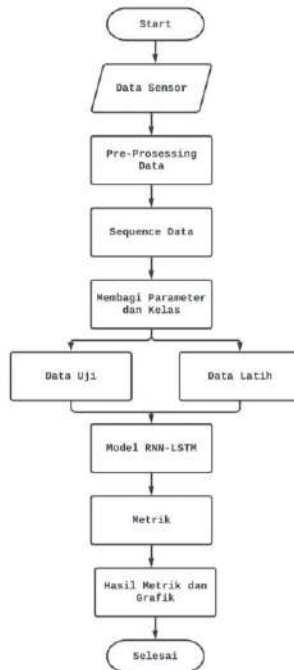


Gambar 18. Flowchart Penggunaan Platform IoT ThingSpeak

Langkah pertama yang dilakukan untuk mengakses *website ThingSpeak* adalah *login* dengan alamat *email* dan kata sandi yang telah didaftarkan agar pengguna dapat mengakses *website*. Jika tidak berhasil, maka proses *login* dilakukan dari awal. Jika berhasil, maka penulis akan diminta untuk membuat *channel* baru, selanjutnya diarahkan ke *channel state* lalu yang dilakukan penulis adalah menghubungkan aplikasi arduino IDE ke *website ThingSpeak*, selanjutnya penulis membuat program dan mengupload program. Jika penguploadan program berhasil pekerjaan dilanjutkan ke pengiriman data, jika tidak pengerjaan diulang ke pembuatan program. Dari grafik jika *chart* sesuai data maka pengerjaan selesai jika tidak sesuai maka pengerjaan diulang ke pembuatan program.

3.1.3.3 Prediksi Data Menggunakan RNN-LSTM

Penerapan model RNN-LSTM terdapat beberapa tahapan yang berpengaruh terhadap hasil akhir dari model tersebut. Setiap tahapan menjadi kunci untuk memastikan ketepatan prediksi yang dihasilkan model.

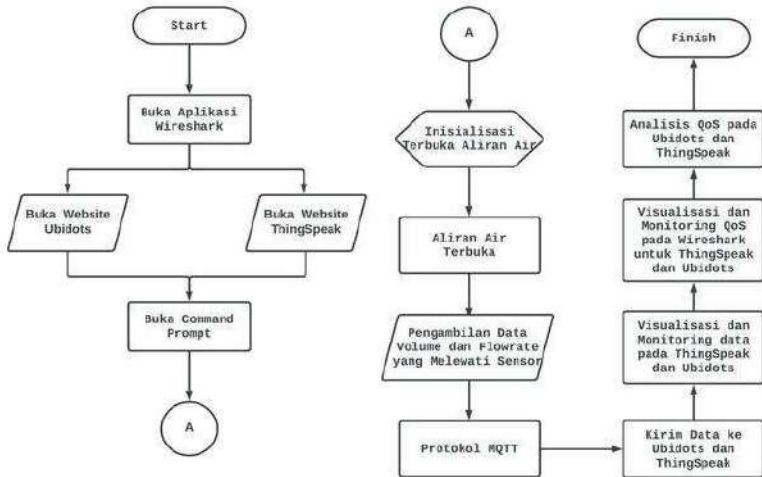


Gambar 19. Flowchart RNN-LSTM

Gambar 19 menjelaskan tahapan dalam memproses dengan model RNN-LSTM. Langkah awal *import dataset* dari hasil sensor YF-S201 yang telah kita dapatkan dan lakukan pelabelan terlebih dahulu secara manual. Lalu, lakukan tahapan *preprocessing* data untuk meningkatkan kualitas dari data, lalu bagi data menjadi data latih dan data uji. Setelah itu menyesuaikan data dengan model RNN-LSTM. Langkah berikutnya, membuat model RNN-LSTM dengan mengatur parameter yang diperlukan untuk memprediksi nilai berdasarkan data. Setelah pengujian, data hasil prediksi di balik dari proses normalisasi untuk mendapatkan nilai semula. Lalu, melakukan perhitungan metrik untuk mengevaluasi kinerja model dengan menggunakan metrik MSE, MAE dan R^2 untuk menilai keakuratan model. Terakhir, ditampilkan hasil metrik dan grafik prediksi untuk memvisualisasikan perbandingan hasil prediksi pada data aktual dan data aplikasi, membantu dalam mengevaluasi sejauh mana model berhasil merepresentasikan pola dalam *sequence* data sensor *water flow* YF-S201.

3.1.3.4 Quality of Service pada Wireshark

Untuk menghitung dan menganalisis QoS, kita harus menggunakan aplikasi *wireshark* sebagai perekam dan menganalisis jaringan data secara *realtime*.



Gambar 20. Flowchart Penggunaan Wireshark

Gambar 20 menjelaskan langkah-langkah untuk menghitung dan menganalisis QoS, pertama kita harus membuka aplikasi *wireshark* serta *Ubidots* dan *ThingSpeak* pada laptop. Lalu, buka *command prompt* pada laptop yang digunakan dan jalankan perintah “ping stem.ubidots.com” dan “ping ThingSpeak.com” untuk

mendapatkan IP *address*. Setelah itu, hubungkan adapter *Mow Flow* ke sumber listrik terdekat, seperti stop kontak. Pastikan *Mow Flow* hidup dan terhubung ke jaringan WiFi yang telah diatur pada perangkat ESP32. Hidupkan stop kran air untuk memulai aliran air melalui *Mowflow*. Amati tampilan grafik *volume* dan *flowrate* pada *Ubidots* dan *ThingSpeak* selama 10 menit. Setelah 10 menit, matikan stop kran air untuk menghentikan aliran air melalui *Mowflow*. Hentikan perekaman di *wireshark*. Selanjutnya, buka hasil perekaman *wireshark*. Terapkan filter pada data *wireshark* sesuai dengan IP *address* yang peroleh dari perintah "ping" pada *command prompt* (`ip.dst == [IP Address Ubidots]` dan `ip.dst == [IP Address ThingSpeak]`). Simpan data yang telah difilter dalam format .CSV, lalu buka file CSV dengan *Microsoft Excel* atau aplikasi *spreadsheet* lainnya. Olah data yang telah didapatkan dari *wireshark* untuk menghitung nilai *throughput*, *jitter*, *delay*, dan *packet loss* menggunakan rumus-rumus yang sudah di tentukan oleh TIPHON dan lakukan analisis data QoS.

3.2 Pengujian

Pengujian adalah langkah penting untuk memastikan bahwa alat dan sistem *website* yang telah dibuat berfungsi dengan baik, sesuai dengan tujuan yang ingin dicapai dan andal dalam penggunaannya. Pengujian mencakup berbagai komponen yaitu sensor *water flow*, *platform IoT Ubidots*, *platform IoT ThingSpeak*, *wireshark* dan RNN-LSTM. Pengujian dilakukan di aliran air di rumah.

3.2.1 Pengujian Alat

Pengujian alat dilakukan sebagai evaluasi terhadap sensor *water flow* yang digunakan untuk memastikan bahwa alat dapat berfungsi dengan baik dan memenuhi kriteria yang ditetapkan.

Tabel 3 menunjukkan pengujian debit air dengan cara manual dimana data yang didapat dari berapa liter air yang dikeluarkan per selama 10 menit menggunakan *stopwatch*. Setelah itu data akan diproses sebagai berikut :

Persamaan perhitungan nilai *error* [36] :

$$Error = \frac{Nilai\ dari\ pembacaan\ alat\ ukur - Nilai\ yang\ dihasilkan\ sensor}{Nilai\ dari\ pembacaan\ alat\ ukur} \times 100 \quad (6)$$

Persamaan perhitungan nilai akurasi :

$$Akurasi = 100\% - Error \quad (7)$$

Tabel 3. Pengujian Sensor *Water Flow*

Perco baan ke-	Flow Manual (L/menit)	Flow Sensor (L/menit)	Error Rate (%)	Akurasi Sensor (%)
1				
2				
3				
4				
...				
8				

3.2.2 Perbandingan Data Aktual dan Prediksi Data Menggunakan RNN-LSTM

Perbandingan RNN-LSTM dengan sensor aliran YF-S201 dapat bermanfaat untuk aplikasi yang memerlukan analisis data aktual dan data aplikasi yang didapat. Dalam menganalisis untuk memprediksi data aktual dan data aplikasi kami mengambil *sampel* data dengan 8 kali pengujian, lalu kami menggunakan model pendekatan RNN-LSTM yang diproses menggunakan *Google Collaboratory*.

Berdasarkan persamaan (1), persamaan (2) dan persamaan (3) yang dimasukkan pada *script program* pada *Google Collaboratory* dihasilkan *codingan* berikut ini untuk memprediksi dan menghitung MSE, MAE dan R-square.

```

# Step 8: Model Testing
y_pred = model.predict(X_test)

# Step 9: Reverse Normalize Data
y_test = scaler.inverse_transform(y_test)
y_pred = scaler.inverse_transform(y_pred)

1/1 [=====] - 0s 336ms/step

# Step 10: Calculating Metrics
mse_actual = mean_squared_error(y_test[:, 0], y_pred[:, 0])
mae_actual = mean_absolute_error(y_test[:, 0], y_pred[:, 0])
r2_actual = r2_score(y_test[:, 0], y_pred[:, 0])

mse_application = mean_squared_error(y_test[:, 1], y_pred[:, 1])
mae_application = mean_absolute_error(y_test[:, 1], y_pred[:, 1])
r2_application = r2_score(y_test[:, 1], y_pred[:, 1])

# Step 11: Showing Results
print("Metrics for 'actual':")
print("Mean Squared Error (MSE):", mse_actual)
print("Mean Absolute Error (MAE):", mae_actual)
print("Coefficient of Determination (R^2):", r2_actual)

print("")

print("Metrics for 'application':")
print("Mean Squared Error (MSE):", mse_application)
print("Mean Absolute Error (MAE):", mae_application)
print("Coefficient of Determination (R^2):", r2_application)

```

Gambar 21. Tampilan Rumus pada Google Colaboratory

3.2.3 Perbandingan Performa QoS (*Quality of Service*) pada *Platform IoT Ubidots* dan *ThingSpeak*

Kami akan menganalisis kriteria jaringan dan menarik kesimpulan dari hasil parameter. Parameter yang digunakan dalam standar TIPHON:

3.2.3.1 *Throughput*

Throughput merujuk pada *bandwidth* yang benar-benar terjadi dalam jangka waktu tertentu ketika mentransmisi file. Meskipun satuan pengukuran yang digunakan sama dengan *bandwidth* (*bit per second/bps*), *throughput* menggambarkan *bandwidth* yang sebenarnya pada saat tertentu dan dalam kondisi serta jaringan spesifik yang digunakan untuk mengunduh file ukuran tertentu. *Throughput* dihitung dengan membagi jumlah total paket yang berhasil dikirimkan ke tujuan selama interval waktu tertentu dengan durasi interval waktu tersebut [36]. Persamaan perhitungan *throughput* [8] :

$$Throughput = \frac{\text{paket data yang diterima}}{\text{lama waktu pengiriman}} \quad (8)$$

Versi TIPHON dari indeks dan kategori *throughput* ditunjukkan pada Tabel 4.

Tabel 4. Kategori *Throughput* [36]

Kategori	<i>Throughput</i>	Indeks
Sangat Baik	>2,1 Mbps	4
Baik	1200 kbps - 2,1 Mbps	3
Cukup	700 - 1200 kbps	2
Kurang Baik	338 - 700 kbps	1
Buruk	0 - 338 kbps	0

3.2.3.2 *Jitter*

Jitter adalah perubahan dalam keterlambatan *end-to-end*. Ketika tingkat *jitter* tinggi terjadi dalam aplikasi berbasis UDP, terutama pada aplikasi *real-time*, hal ini dapat menciptakan situasi yang tidak dapat diterima. *Jitter* dalam kasus tersebut dapat mengakibatkan distorsi dalam sinyal, dan satu-satunya cara untuk memperbaikinya adalah dengan meningkatkan *buffer* dalam antrean [37]. Persamaan perhitungan *jitter* [8].

$$Jitter = \frac{\text{total variasi delay}}{\text{total paket data yang diterima} - 1} \quad (9)$$

Versi TIPHON dari indeks dan kategori *jitter* ditunjukkan pada Tabel 6.

Tabel 5. Kategori *Jitter* [36]

Kategori	<i>Jitter</i>	Indeks
Sangat Baik	0 ms	4
Baik	0 ms - 75 ms	3
Cukup	75 ms - 125 ms	2

Buruk	125 ms - 225 ms	1
-------	-----------------	---

3.2.3.3 Delay

Delay adalah waktu yang diperlukan untuk data untuk paket yang dikirim dari satu komputer ke komputer tujuan. *Delay* dalam proses pengiriman paket dalam jaringan komputer dapat disebabkan oleh antrian yang panjang atau pengambilan rute alternatif untuk menghindari kemacetan *routing*. Latensi dapat dipengaruhi oleh jarak, kendaraan fisik, penyumbatan atau waktu pemrosesan yang lama. Untuk mengetahui *delay* pada paket ditransmisikan, dapat dilakukan dengan membagi panjang paket (dalam satuan bit) dengan *bandwidth link* (dalam satuan *bit/s*) [36]. Persamaan perhitungan nilai *delay* [8] :

$$\text{Rata - rata delay} = \frac{\text{total delay}}{\text{paket data yang diterima}} \quad (10)$$

Versi TIPHON dari indeks dan kategori *delay* ditunjukkan pada Tabel 7.

Tabel 6. Kategori Delay [36]

Kategori	Delay	Indeks
Sangat Baik	<150 ms	4
Baik	150 – 300 ms	3
Cukup	300 – 450 ms	2
Buruk	>450ms	1

3.2.3.4 Packet Loss

Packet loss adalah persentase paket yang hilang selama proses pengiriman data. Hal ini disebabkan berbagai faktor, seperti penurunan sinyal dalam media jaringan, kegagalan perangkat keras jaringan atau radiasi dari lingkungan sekitarnya. *Packet loss* menggambarkan kondisi dimana jumlah total paket yang hilang, yang bisa terjadi akibat tabrakan dan kemacetan jaringan. Hal ini mempengaruhi semua aplikasi karena retransmisi akan mengurangi efisiensi jaringan secara keseluruhan, meskipun *bandwidth* yang cukup tersedia untuk aplikasi tersebut [36]. Persamaan perhitungan nilai *packet loss* [8]:

$$\text{Packet Loss} = \frac{\text{paket data yang dikirim} - \text{paket data yang diterima}}{\text{paket data yang diterima}} \times 100\% \quad (11)$$

Versi TIPHON dari indeks dan kategori *packet loss* ditunjukkan pada Tabel 5.

Tabel 7. Kategori Packet Loss [36]

Kategori	Packet Loss	Indeks
Sangat Baik	0 – 2%	4
Baik	3 – 14%	3
Cukup	15 – 24%	2
Buruk	>25%	1

Pengujian QoS dilakukan untuk menentukan performansi kinerja jaringan. Pengujian QoS menggunakan *software* Arduino IDE dan *wireshark* sebagai sistem. Dalam pengujian QoS, dilakukan 8 pengujian dengan waktu 10 menit setiap pengujian. Masing-masing tes menghasilkan *throughput*, *packet loss*, *jitter* dan *delay* dengan tabel perbandingan berikut:

Tabel 8. Metode Tabel Perbandingan Performa QoS pada Platform IoT Ubidots dan ThingSpeak

Pengujian ke-	Ubidots				ThingSpeak			
	Throughput (bps)	Delay (ms)	Jitter (ms)	Packet Loss (%)	Throughput (kbps)	Delay (ms)	Jitter (ms)	Packet Loss (%)
1								
2								
3								
4								
...								
8								

Bab 4. Hasil dan Pembahasan

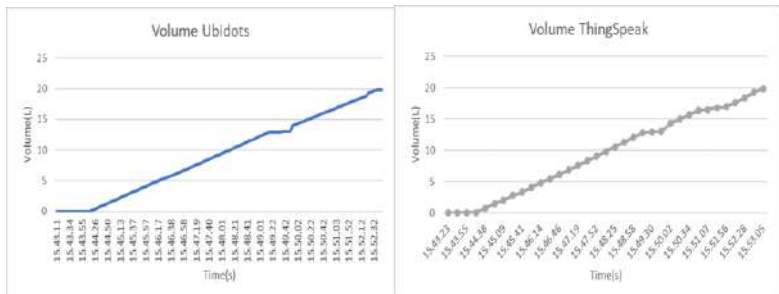
Hasil dan pembahasan mencakup berbagai aspek pengujian yaitu pengujian *mow flow*, pengujian data aktual dan prediksi data menggunakan RNN-LSTM dan perbandingan performa QoS pada *platform IoT Ubidots* dan *platform IoT ThingSpeak*.

4.1. Pengujian Alat

4.1.1 Pengujian Volume

Selama proses pengujian volume pada *platform Ubidots* dan *ThingSpeak*, kami telah menjalankan 8 kali percobaan, dengan setiap percobaan berlangsung selama 10 menit, sebagaimana terlihat pada Gambar 22 - Gambar 29. Pada gambar tersebut merupakan penyajian data dalam durasi 10 menit, namun grafik yang disajikan dibuat rapat karena kurangnya *space*.

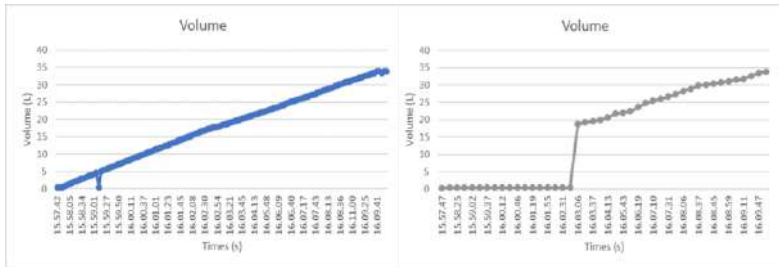
4.1.1.1 Pengujian 1



Gambar 22. Pengujian 1 Grafik Volume

Pada pengujian satu data yang diperoleh oleh *Ubidots* dan *ThingSpeak* memiliki hasil volume yang sama di angka 19.84 L pada waktu awal pengiriman dan waktu akhir pengiriman data yang berbeda, *website Ubidots* lebih unggul dalam *interval rate* yang lebih rapat dibandingkan *ThingSpeak*.

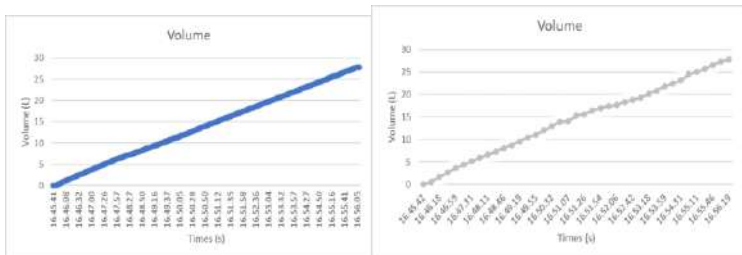
4.1.1.2 Pengujian 2



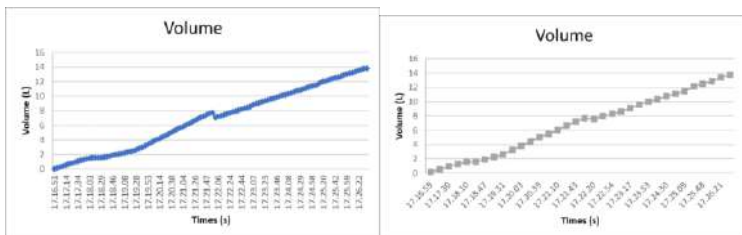
Gambar 23. Pengujian 2 Grafik Volume

Pada pengujian 2, terdapat kendala pada jaringan yang menyebabkan *ThingSpeak* baru mengirimkan data pada pukul 16:03:06. Sementara itu, pada *Ubidots*, data terkirim secara konsisten, namun sempat terputus pada pukul 15:59:01, yang mengakibatkan data terbaca 0 setelah itu *Ubidots* kembali mengirimkan datanya kembali.

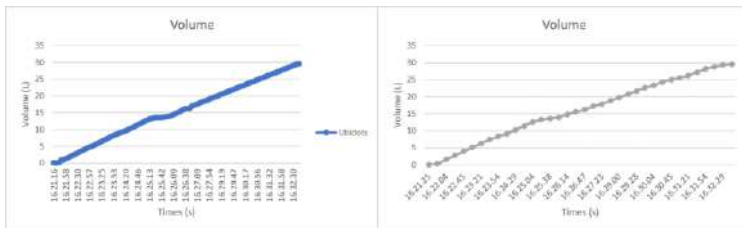
4.1.1.3 Pengujian 3, Pengujian 4, Pengujian 5 dan Pengujian 6



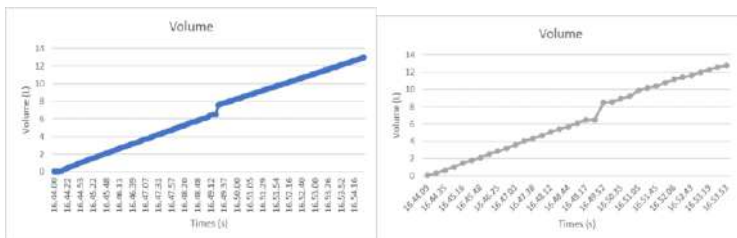
Gambar 24. Pengujian 3 Grafik Volume



Gambar 25. Pengujian 4 Grafik Volume



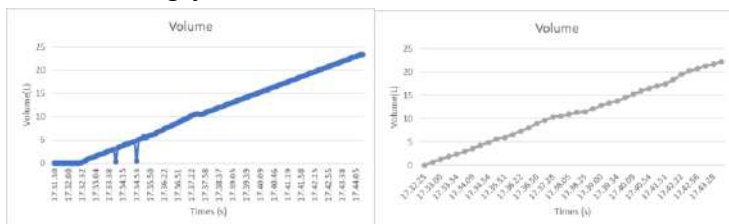
Gambar 26. Pengujian 5 Grafik Volume



Gambar 27. Pengujian 6 Grafik Volume

Pada Gambar 24, 25, 26 dan 27 pola pengiriman data ke *Ubidots* dan *ThingSpeak* menunjukkan kemiripan dalam *volume* akhir. Pada Gambar 24, *volume* mencapai 27.8 liter. Pada Gambar 25, *volume* turun menjadi 13.8 liter. Gambar 26 menunjukkan *volume* mencapai 30 liter, sementara pada Gambar 27, *volume* mencapai 13 liter. Pola ini menandakan variasi yang signifikan dalam jumlah *volume* yang terdeteksi, dengan perubahan yang terlihat pada setiap pengambilan data.

4.1.1.4 Pengujian 7

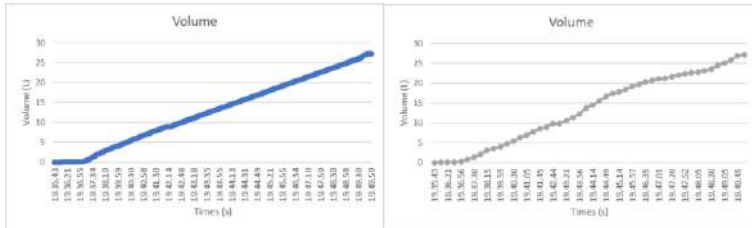


Gambar 28. Pengujian 7 Grafik Volume

Pada Gambar 28, terdapat perbedaan data pada awal karena *Ubidots* mengalami 2 kali kehilangan data, yang disebabkan oleh kehilangan koneksi pada

pukul 17:34:01 dan pukul 17:34:53. Namun, setelah itu, *Ubidots* mampu mengirimkan data dengan konsisten yang sesuai.

4.1.1.5 Pengujian 8



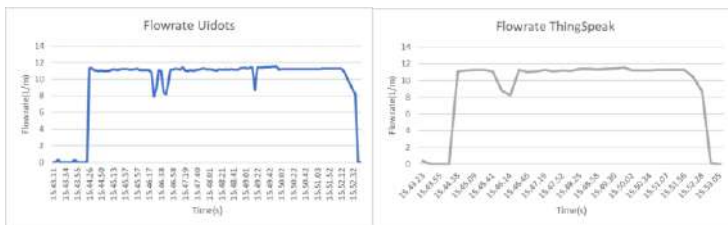
Gambar 29. Pengujian 8 Grafik Volume

Pada Gambar 29, terlihat bahwa data yang dikirimkan *Ubidots* dan *ThingSpeak* memiliki pola yang sama, dimana *volume* data selalu stabil sesuai dengan air yang mengalir. Kondisi ini memperlihatkan bahwa meskipun terdapat perbedaan waktu dalam pengiriman data, namun hasil pola pengiriman data *volume* tetap terjaga.

4.1.2 Pengujian Flow Rate

Selama proses pengujian volume pada *platform Ubidots* dan *ThingSpeak*, kami telah menjalankan 8 kali percobaan, dengan setiap percobaan berlangsung selama 10 menit, sebagaimana terlihat pada Gambar 30 - Gambar 37. Setelah melakukan pengujian, penulis mendapatkan *Ubidots* lebih unggul dalam penerimaan data dibandingkan *Thingspeak*, dimana *Ubidots* menerima 1 data perdetiknya dengan rata-rata *delay* dalam semua pengujian sebesar 5 detik. Sedangkan *ThingSpeak* menerima 1 data per 15 detik dengan rata-rata *delay* pada semua pengujian sebesar 17 detik.

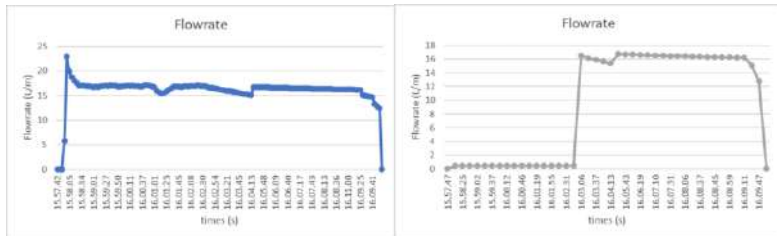
4.1.2.1 Pengujian 1



Gambar 30. Pengujian 1 Grafik Flow Rate

Pada Gambar 30, kedua *platform* menunjukkan pola yang serupa, tetapi terdapat perbedaan. *Ubidots* menampilkan 3 lembah, sementara *ThingSpeak* hanya menampilkan 1 lembah. Perbedaan ini disebabkan oleh adanya delay yang berbeda pada masing-masing *platform*. Meskipun pola umumnya serupa, variasi *delay* menghasilkan perbedaan jumlah dan *interval* lembah yang teramati di kedua *platform*.

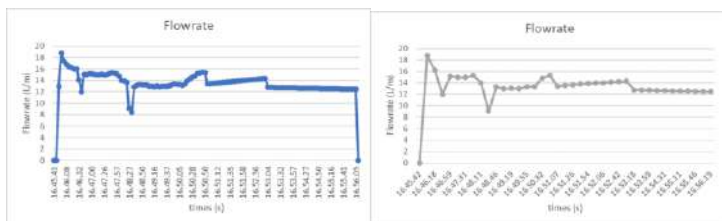
4.1.2.2 Pengujian 2



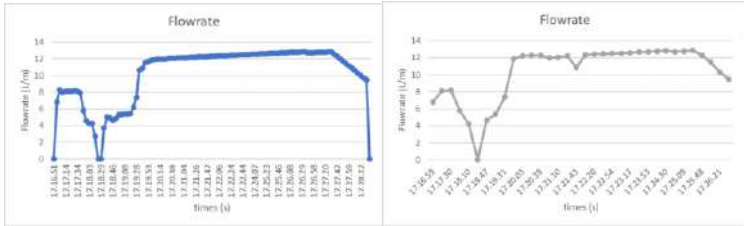
Gambar 31. Pengujian 2 Grafik Flow Rate

Gambar 31, terjadi putus koneksi ke *server* pada *platform ThingSpeak*, mengakibatkan pengiriman data dengan nilai nol. Hal ini disebabkan oleh lambatnya proses penangkapan jaringan. Ketika koneksi kembali terhubung ke *server*, *ThingSpeak* merespon dengan baik dengan mengirimkan data terakhir yang telah disimpan sebelum putus koneksi. Proses ini menunjukkan kemampuan *ThingSpeak* untuk mengatasi kendala koneksi.

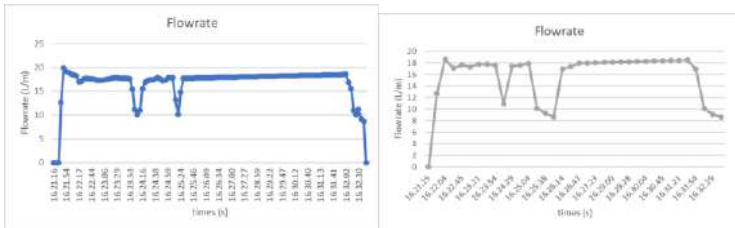
4.1.2.3 Pengujian 3, Pengujian 4, Pengujian 5, Pengujian 6, dan Pengujian 7



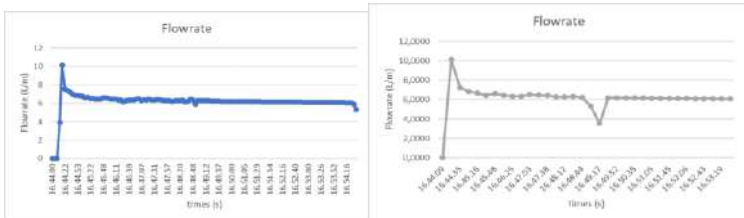
Gambar 32. Pengujian 3 Grafik Flow Rate



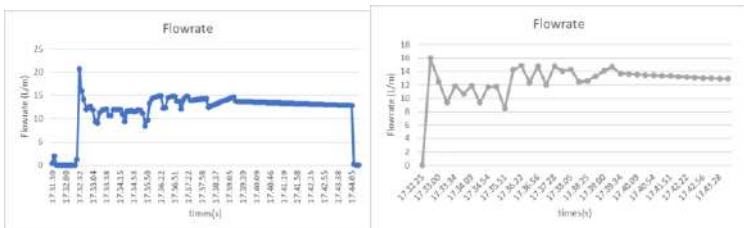
Gambar 33. Pengujian 4 Grafik Flow Rate



Gambar 34. Pengujian 5 Grafik Flow Rate



Gambar 35. Pengujian 6 Grafik Flow Rate

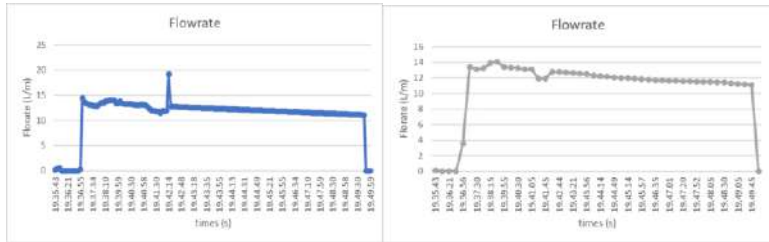


Gambar 36. Pengujian 7 Grafik Flow Rate

Pola pengiriman *Ubidots* dan *ThingSpeak* serupa pada Gambar 32, 33, 34, 35, dan 36. Namun, ada perbedaan saat *Ubidots* telah selesai melakukan

pengujian, *platform* ini mengirim data dengan nilai nol. Sementara itu, *ThingSpeak* tetap mempertahankan titik terakhir hingga data baru berhasil dikirim. Perbedaan ini menunjukkan cara kedua *platform* menangani pengiriman data, di mana *ThingSpeak* lebih memilih untuk mempertahankan nilai terakhir selama tidak ada pembaruan data baru.

4.1.2.4 Pengujian 8



Gambar 37. Pengujian 8 Grafik Flow Rate

Gambar 37 menunjukkan pola pengiriman data yang hampir sama antara *platform Ubidots* dan *ThingSpeak*. Namun, pada pukul 19:42:14 terjadi perbedaan yang mencolok, dimana *Ubidots* menunjukkan lonjakan data, sedangkan *ThingSpeak* tidak merekam lonjakan tersebut. Perbedaan ini disebabkan *delay* yang berbeda pada kedua *platform*, yang mempengaruhi kemampuan *ThingSpeak* untuk merekam lonjakan data pada waktu yang sama dengan *Ubidots*.

4.1.3 Error dan Akurasi

Tabel 9. Pengujian Sensor Water Flow

Percobaan ke-	Flow Manual (L/menit)	Flow Sensor (L/menit)	Error Rate (%)	Akurasi Sensor (%)
1	20	19.84	0.8%	99.20%
2	34.4	33.85	1.6%	98.40%
3	28	27.76	0.86%	99.14%
4	14	12.77	1.64%	98.36%
5	30	29.54	1.53%	98.47%
6	13	12.96	0.31%	99.69%

7	23.8	23.33	1.97%	98.03%
8	28	27.21	2.82%	97.18%
Rata-Rata			1.44%	98.56%

Berdasarkan data pengukuran yang sudah dilakukan, dapat disimpulkan bahwa sensor memiliki akurasi yang baik dengan kesalahan pengukuran rata-rata 1.44% dan memiliki akurasi sebesar 98.56%. Hal ini menunjukkan bahwa sensor dapat mengirimkan data ke *platform IoT Ubidots* dan *ThingSpeak* secara baik dan dapat diandalkan dalam melakukan pengukuran. Meskipun masih ada variasi dalam kesalahan antara pengamatan yang berbeda. Namun secara umum, aplikasi ini memberikan hasil yang akurat.

4.2. Perbandingan Data Aktual dan Prediksi Data Menggunakan RNN-LSTM

Kami melakukan evaluasi kinerja model RNN-LSTM menggunakan 3 metrik yaitu MSE, MAE dan *R-square*. Perhitungan parameter tersebut diperoleh dengan memanggil fungsi *mean_squared_error*, *mean_absolute_error* dan *r2_score* dari *library Scikit-Learn* sesuai dengan persamaan (1), persamaan (2) dan persamaan (3). Persamaan tersebut menjadi inputan dari *running script* program yang dapat dilihat pada Gambar 21, dimana hasil metrik dapat dilihat pada Gambar 38.

```

Metrics for 'actual':
Mean Squared Error (MSE): 1.2824408576826671
Mean Absolute Error (MAE): 1.0128360748291012
Coefficient of Determination (R^2): 0.709197084425699

Metrics for 'application':
Mean Squared Error (MSE): 1.197032793162859
Mean Absolute Error (MAE): 0.94859832763672
Coefficient of Determination (R^2): 0.6819447355821933

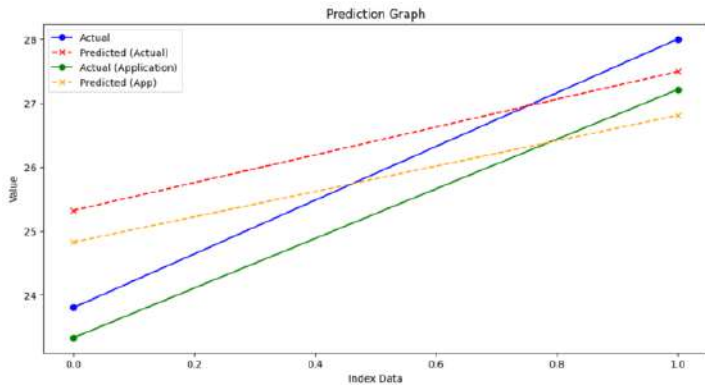
```

Gambar 38. Hasil Perhitungan Matrik

Nilai MSE pada data aktual adalah 1.28 dan data aplikasi sebesar 1.19. Pada nilai MAE data aktual adalah 1.01 dan data aplikasi sebesar 0.94. Semakin kecil nilai MSE dan MAE berarti semakin akurat model dalam memprediksi pola kedua data dengan error yang kecil. Dengan kata lain, akurasi prediksi model terhadap target cukup tinggi. Nilai *R-square* data aktual sebesar 0.709 dan data aplikasi sebesar 0.681 mendekati 1. Hal ini berarti kecocokan antara data yang diproses

dan hasil prediksi model cukup baik. Artinya kontribusi model dalam menjelaskan variasi target cukup besar. Hal ini sesuai dengan dengan penelitian sebelumnya oleh Badriyah et al. [38].

Dengan nilai matrik yang menunjukkan performa prediksi yang baik, dapat disimpulkan bahwa model RNN-LSTM yang diterapkan pada penelitian ini layak dan akurat digunakan untuk memprediksi data aktual dan data aplikasi berdasarkan variabel input *hour*.



Gambar 39. Grafik Data Prediksi dan Data Aktual

Tujuan dari grafik prediksi ini adalah untuk mengevaluasi secara visual performa model prediksi, yaitu seberapa akurat model memprediksi nilai aktual berdasarkan pola *sequence* data waktu.

Gambar 39 menunjukkan grafik prediksi dengan sumbu X berfungsi untuk merepresentasikan urutan data *time series* (waktu). Sedangkan sumbu Y menunjukkan nilai aktual dan nilai hasil prediksi. Dari grafik ini dapat dianalisis bahwa tingkat akurasi prediksi model RNN-LSTM paling optimal ditunjukkan pada indeks 0, dimana nilai aktual dan nilai prediksi saling berhimpit. Hal ini mengindikasikan akurasi prediksi model sangat baik pada data testing pertama. Tingkat akurasi yang optimal ini juga sesuai dengan nilai R-square sekitar 0,7 yang mendekati 1 untuk kedua target data. Dengan demikian, performa prediksi model RNN-LSTM terbaik ditunjukkan pada indeks 0 grafik dimana nilai aktual dan hasil prediksi saling berhimpit.

4.3 Perbandingan Performa QoS (*Quality of Service*) pada Platform IoT Ubidots dan ThingSpeak

Hasil pengukuran parameter *throughput*, *jitter*, *delay* dan *packet loss* diuji menggunakan *software wireshark* dan menggunakan *server* yang sama. Semua

parameter diuji berdasarkan data yang diperoleh dari *platform* IoT *Ubidots* dan *platform* IoT *ThingSpeak* dengan pengamatan selama 10 menit dalam 8 percobaan.

4.3.1 Hasil Pembacaan *Throughput*

Statistics		
Measurement	Captured	Displayed
Packets	2188	401 (18.3%)
Time span, s	351.329	350.773
Average pps	6.2	1.1
Average packet size, B	183	76
Bytes	400573	30415 (7.6%)
Average bytes/s	1140	86
Average bits/s	9121	693

Gambar 40. Perhitungan pada *Throughput*

Persamaan (8) dapat digunakan untuk mengetahui dan menghitung hasil perhitungan *throughput* berdasarkan data yang ditunjukkan pada gambar di atas:

$$Throughput = \frac{30415 \text{ Bytes}}{350.773 \text{ s}} = 86.70849 \text{ bytes/s} \times 8 = 693.67 \text{ bps}$$

$$Throughput = \frac{693.67 \text{ bps}}{1000} = 0.69367 \text{ Kbps}$$

Hasil pengukuran *throughput*, yang dihitung berdasarkan data yang telah diperoleh menghasilkan data sebagai berikut



Gambar 41. Grafik Perbandingan *Throughput*

Berdasarkan Gambar 41. ditampilkan grafik rata-rata *throughput* yang dihasilkan *Ubidots* sebesar 0.68139 kbps, sedangkan *ThingSpeak* sebesar 0.70238 kbps. Hasil Dari hasil ini, dapat disimpulkan bahwa kinerja kedua jaringan ini secara umum dapat dianggap buruk dan sesuai dengan standar yang ditetapkan oleh TIPHON.

4.3.2 Hasil Pembacaan *Jitter*

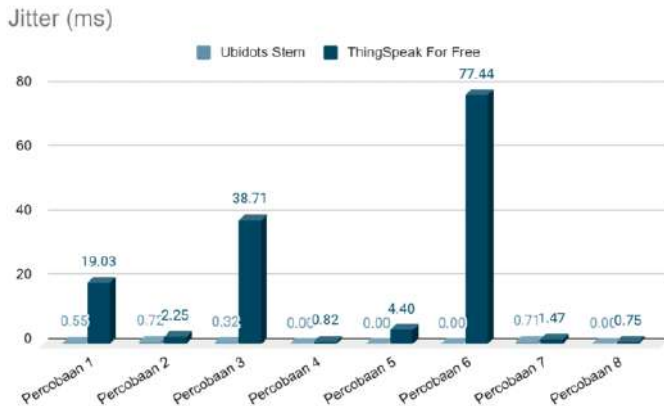
Statistics		
Measurement	Captured	Displayed
Packets	2188	401 (18.3%)
Time span, s	351.329	350.773
Average pps	6.2	1.1
Average packet size, B	183	76
Bytes	400573	30415 (7.6%)
Average bytes/s	1140	86
Average bits/s	9121	693

Gambar 42. Perhitungan pada *Jitter*

Berdasarkan data pada gambar 42, nilai *jitter* dapat dihitung dengan menggunakan persamaan (9).

$$Jitter = \frac{0.21986 s}{401 \text{ packet} - 1} = 0.000548 s \times 1000 = 0.55 \text{ ms}$$

Hasil pengukuran *jitter* berdasarkan data yang diperoleh hasil sebagai berikut.



Gambar 43. Grafik Perbandingan *Jitter*

Gambar 43 menunjukkan bahwa grafik rata-rata *jitter* yang dihasilkan oleh *Ubidots* adalah 0.29 ms, sementara *ThingSpeak* mencapai 18.11 ms. Dari hasil ini,

dapat disimpulkan bahwa jaringan yang menunjukkan kinerja kedua *platform* ini di index baik, namun yang lebih baik adalah jaringan *Ubidots* hal ini sesuai dengan standar yang ditetapkan oleh TIPHON.

4.3.3 Hasil Pembacaan *Delay*

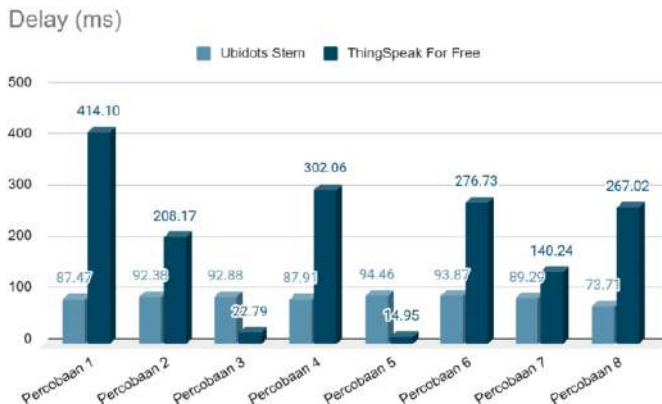
Statistics		
Measurement	Captured	Displayed
Packets	2188	401 (18.3%)
Time span, s	351.329	350.773
Average pps	6.2	1.1
Average packet size, B	183	76
Bytes	400573	30415 (7.6%)
Average bytes/s	1140	86
Average bits/s	9121	693

Gambar 44. Perhitungan pada *Delay*

Berdasarkan data pada gambar 44, nilai *delay* dapat dihitung dengan menggunakan persamaan (10)

$$Total\ Delay = \frac{350.773\ s}{401\ packet} = 0,8747\ s \times 1000 = 87.47\ ms$$

Data berikut dihasilkan dari pengukuran *delay* yang dihitung berdasarkan data yang telah diperoleh hasil sebagai berikut.



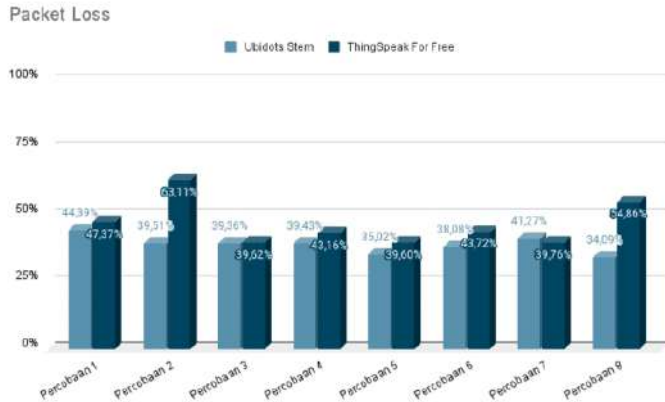
Gambar 45. Grafik Perbandingan *Delay*

Grafik pada Gambar 45 menunjukkan bahwa rata-rata *delay* yang dihasilkan oleh *Ubidots* adalah 89 ms, sementara *ThingSpeak* mencapai 205.76 ms. Dari data tersebut, dapat disimpulkan bahwa kinerja jaringan yang lebih baik adalah milik

Ubidots, sedangkan *ThingSpeak* masih di indeks baik yang sesuai dengan standar yang ditetapkan oleh TIPHON.

4.3.4 Hasil Pembacaan *Packet Loss*

Hasil pengukuran *packet loss*, yang dihitung berdasarkan persamaan (11) yang telah diperoleh menghasilkan data sebagai berikut.



Gambar 46. Grafik Perbandingan *Packet Loss*

Pada Gambar 46. terlihat persentase *packet loss* cukup bervariasi dan tinggi pada platform IoT *Ubidots* dan *ThingSpeak*, dengan rata-rata *packet loss Ubidots* sebesar 38.89% dan *packet loss Thingspeak* sebesar 46.40%. Menurut standar TIPHON, jika *packet loss* melebihi 25% maka kualitas jaringan dikategorikan buruk. Tingginya angka *packet loss* dikarenakan *interval rate* yang dikirim oleh kedua platform IoT, sehingga menyebabkan *buffering* yang *overload*. Akibatnya sebagian data gagal ditampilkan di *Ubidots* dan *ThingSpeak*.

Dengan kata lain, tingginya *packet loss* pada penelitian ini bukan hanya dipengaruhi oleh masalah konektivitas jaringan, melainkan juga terbatasnya kapasitas *buffer* pada server *Ubidots* dan *Thingspeak*.

4.4. Pembahasan

Proses pengiriman data ketika sensor YF-S201 membaca *volume* dan *flow rate* menggunakan platform IoT *Ubidots* dapat dianggap sebagai sistem IoT yang baik dan tingkat pengiriman data yang lebih cepat dibanding dengan platform IoT *ThingSpeak*. Hal ini sesuai dengan penelitian sebelumnya oleh K. A. Pranoto et al. [39] yang menunjukkan bahwa data lebih cepat dengan rata-rata *delay* pada setiap pengujian 5 detik dibanding dengan platform IoT *ThingSpeak* dengan rata-rata *delay* pada setiap pengujian 17 detik.

Untuk menentukan semua parameter QoS, pengguna memiliki empat jenis indeks yaitu sangat baik, baik, cukup dan buruk dapat dilihat dalam Tabel 10. Nilai rata-rata atau persentase dari masing-masing parameter QoS dapat digunakan untuk mengetahui penentuan indeks tersebut.

Tabel 10. Standar Persentase dan Nilai QoS [31]

Indeks	Persentase (%)	Nilai
Sangat Baik	95 - 100	3.8 - 4
Baik	75 - 94.75	3 - 3.79
Cukup	50 - 74.75	2 - 2.99
Buruk	25 - 49.75	1 - 1.99

Berdasarkan data yang tercatat dalam Tabel 11, terlihat bahwa rata-rata nilai pada setiap parameter QoS yang diukur selama 10 menit dalam 8 kali pengujian menunjukkan hasil yang cukup baik untuk kedua *platform* IoT. Rata-rata *throughput* untuk *Ubidots* mencapai 0.68139 kbps, sedangkan *ThingSpeak* mencapai 0.61556 kbps. Pada parameter *jitter*, keduanya dapat dikategorikan baik dengan nilai rata-rata 0.29 ms untuk *Ubidots* dan 18,11 ms untuk *ThingSpeak*. *Ubidots* menunjukkan kinerja yang sangat baik pada parameter *delay* dengan rata-rata 89 ms, sementara *ThingSpeak* dikategorikan baik dengan rata-rata sebesar 205.76 ms. Pada parameter *packet loss* menunjukkan hasil yang buruk dengan rata-rata 38.89% untuk *platform* lot *Ubidots* dan 46.40% untuk *platform* IoT *ThingSpeak*.

Tabel 11. Rata-Rata Parameter QoS pada *Ubidots* dan *ThingSpeak*

Parameter	<i>Ubidots</i>			<i>ThingSpeak</i>		
	Rata-rata Nilai	Indeks	Kategori	Rata-rata Nilai	Indeks	Kategori
<i>Throughput (kbps)</i>	0.68139	0	BK	0.61556	0	BK

Jitter (ms)	0.29	3	B	18.11	3	B
Delay (ms)	89	4	SB	205.76	3	B
Packet Loss (%)	38.89%	1	B	46.40%	1	B
Rata - rata Indeks		2	CUKUP		1.75	BURUK

Indeks total rata-rata *Ubidots* mencapai 2 dengan kategori cukup, sementara *ThingSpeak* memperoleh hasil 1.75 dikategori buruk berdasarkan standar persentase TIPHON.

Sebagai hasil dari perbandingan nilai parameter QoS rata-rata, dapat disimpulkan bahwa layanan internet *Ubidots* menunjukkan performa yang lebih baik daripada *ThingSpeak*. Penggunaan parameter QoS (*throughput*, *jitter*, *delay*, dan *packet loss*) terbukti efektif dalam menganalisis kinerja suatu layanan internet. Penting juga untuk diperhatikan bahwa selain faktor QoS, faktor eksternal seperti kondisi cuaca, kebisingan dan lokasi juga dapat mempengaruhi kualitas QoS.

Pengujian ini menetapkan *update interval* untuk kode pemrograman ESP32 pada kedua *platform* IoT, yaitu 1000 ms atau 1 detik. Menurut kami hal ini sudah optimal, karena dapat memastikan *update dashboard* secara *real-time* tanpa melebihi batas *rate limit* yang sudah ditetapkan oleh masing-masing *platform*. *Ubidots Stem* memiliki *rate limit* 1 request/detik [40], sedangkan *ThingSpeak* yang kami gunakan (*For Free*) memiliki *rate limit* 1 request/ 15 detik [41]. Dengan *update interval* 1 detik, kami dapat memastikan bahwa kedua *server platform* IoT tidak akan terbebani oleh permintaan yang berlebihan.

Selain itu, *server* yang digunakan pada *Ubidots* dan *ThingSpeak* sama yaitu *Starhub Ltd* Singapore. Hal ini dilakukan untuk memastikan bahwa kondisi jaringan internet yang digunakan sama. Sehingga perbedaan performa *Quality of Service* yang terukur murni disebabkan oleh perbedaan antara *platform Ubidots* dan *ThingSpeak*.

Dalam membandingkan kinerja *Ubidots* dan *ThingSpeak*, hasil pengujian QoS yang dilakukan menjadi lebih akurat dan objektif dengan menyamakan parameter-parameter penting ini.

Model RNN-LSTM yang digunakan dalam penelitian ini dimulai dengan membuat *sequence* data yang akan digunakan sebagai *input* untuk model. Setiap baris dalam data tersebut mencakup dua nilai '*hour*' sebagai *input* (X) dan nilai '*actual*' dan '*application*' sebagai *output* (Y). Model RNN-LSTM kemudian dibangun menggunakan *sequential* API Keras dengan satu *layer* LSTM dengan 64 *neuron* dan fungsi aktivasi *relu*, diikuti oleh satu *layer Dense* dengan 2 *neuron* sebagai *output*. Model dikompilasi dengan *optimizer Adam* dan fungsi *loss Mean Squared Error* (MSE), lalu dilatih selama 500 *epochs* dan *batch size* 32.

Setelah proses pelatihan, model dievaluasi pada data *testing*. Nilai prediksi di balik-normalisasi agar skalanya sesuai dengan data *actual*. Evaluasi dilakukan dengan metrik regresi *Mean Squared Error* (MSE), *Mean Absolute Error* (MAE), dan *Coefficient of Determination* (R^2) baik pada target '*actual*' maupun '*application*'. Hasil evaluasi metrik digunakan untuk menilai seberapa baik model mampu memprediksi kedua target berdasarkan pola data *time series* jam yang diberikan sebagai *input*.

Tabel 12. Parameter pada RNN-LSTM

Jenis	Nilai/Uraian
LSTM Layer	64
Activation	relu
Optimizer	Adam
Epoch	500
Batch Size	32

Pada model RNN-LSTM yang kami buat dalam penelitian ini, tidak ada *hidden layer* tambahan. *Layer* LSTM sudah dapat memodelkan dependensi waktu dan memiliki kemampuan untuk menangkap pola jangka panjang dalam *sequence* data. *Layer* LSTM sudah berperan sebagai *hidden layer*. Hal ini sesuai dengan penelitian sebelumnya yang menjelaskan bahwa RNN-LSTM dapat secara otomatis mengekstrak fitur dari data *time series* dan mengubahnya menjadi representasi level tinggi tanpa memerlukan *hidden layer* tambahan [42].

Bab 5. Kesimpulan dan Saran

5.1. Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan dapat ditarik kesimpulan bahwa :

1. Sistem monitoring aliran air pada rumah pintar dirancang dengan mengintegrasikan sensor *water flow* YF-S201 ke mikrokontroler NodeMCU ESP32, yang dihubungkan dengan *platform* IoT *Ubidots* dan *ThingSpeak* melalui jaringan WiFi. Data sensor dikirim dan ditampilkan secara *real time* di kedua *platform* IoT.
2. Performansi pengiriman data sensor *Ubidots* terbukti lebih unggul dalam *request* data dengan *delay* keseluruhan pengiriman sebesar 5 detik sedangkan *ThingSpeak* memiliki rata-rata *delay* sebesar 17 detik serta *Ubidots* memiliki tampilan data yang lebih informatif dalam *dashboard*-nya dibandingkan *ThingSpeak*.
3. Performa prediksi model RNN-LSTM terhadap data aktual dan data aplikasi optimal, ditunjukkan oleh nilai MSE yaitu 1.28 untuk data aktual dan 1.19 untuk data aplikasi. Nilai MAE yaitu 1.01 untuk data aktual dan 0.94 untuk data aplikasi, serta nilai *R-square* cukup tinggi mendekati 1, yaitu 0.709 untuk data aktual dan 0.681 untuk data aplikasi. Hal ini menunjukkan akurasi prediksi model RNN-LSTM dalam memprediksi target aktual dan aplikasi berdasarkan pola *sequence* data jam.
4. Evaluasi kinerja jaringan IoT sangat penting untuk pengukuran dan analisis QoS *metric*. Kami telah menunjukkan bahwa *platform Ubidots* memiliki keunggulan pada parameter *jitter* dan *delay* pada *metric* ini.
5. Sensor *water flow* YF-S201 mampu menunjukkan hasil akurasi pengukuran yang sangat baik dengan rata-rata error 1.44%. Ini membuktikan bahwa data sensor yang dihasilkan dapat diandalkan untuk *memonitoring* air.

5.2. Saran

Untuk menyelesaikan penelitian akhir ini, masih terdapat banyak aspek yang perlu di kembangkan lebih lanjut. Oleh karena itu, penulis ingin memberikan beberapa saran sebagai berikut :

1. Melakukan pengujian dengan rentang waktu yang lebih panjang untuk mendapatkan gambaran performa *Ubidots* dan *ThingSpeak* serta kualitas jaringan internet yang lebih baik.
2. Menggunakan opsi *ThingSpeak* berbayar untuk mendapatkan performa yang optimal dan memastikan kesamaan *interval rate* dengan *Ubidots* yaitu 1 *request*/detik.

3. Meningkatkan ukuran data latih pada model RNN-LSTM agar performa prediksi lebih akurat.
4. Melakukan analisis pengaruh faktor eksternal terhadap parameter QoS secara kuantitatif, seperti melakukan pengujian ulang pada lokasi dan lingkungan yang berbeda.
5. Melakukan penelitian lanjutan dengan membandingkan *platform* IoT lainnya dengan *Ubidots* dan *ThingSpeak*.

Daftar Pustaka

- [1] A. B. Ramadhan, S. Sumaryo, and R. A. Priramadhi, "Desain dan Implementasi Pengukuran Debit Air Menggunakan Sensor Water Flow Berbasis IoT," *eProceedings of Engineering*, vol. 6, no. 2, Aug. 2019.
- [2] S. R. Siregar and R. Widyasari, "Peramalan Harga Crude Oil Menggunakan Metode Long Short-Term Memory (LSTM) dalam Recurrent Neural Network (RNN)," *Jurnal Lebesgue : Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika*, vol. 4, no. 3, pp. 1478–1489, Sep. 2023, doi: 10.46306/lb.v4i3.421.
- [3] A. Arifin and M. Rizal, "Implementasi Sistem Otomatisasi Perawatan Tanaman Indoor Berbasis Internet of Things (IoT)," *remik*, vol. 7, no. 2, pp. 935–945, Apr. 2023, doi: 10.33395/remik.v7i2.12277.
- [4] "Sistem Monitoring dan Notifikasi Kualitas Udara dalam Ruangannya dengan Platform IoT." <https://cogito.unklab.ac.id/index.php/cogito/article/view/105/74> (accessed Jun. 27, 2023).
- [5] A. T. Putra and R. Risfendra, "Penggunaan Aplikasi Ubidots untuk Sistem Kontrol dan Monitoring pada Gudang Gula Berbasis Arduino UNO," *JTEIN: Jurnal Teknik Elektro Indonesia*, vol. 2, no. 1, pp. 40–48, Feb. 2021, doi: 10.24036/jtein.v2i1.120.
- [6] M. Ikhwanusshofa, A. Nuramal, and N. I. Supardi, "Pemanfaatan Internet Of Things Untuk Monitoring Suhu di BPPT–MEPPO," *Rekayasa Mekanika*, vol. 4, no. 1, pp. 19–24, Oct. 2020, doi: 10.33369/rekayasamekanika.v4i1.13274.
- [7] R. Rusli and A. Azizah, "Analisis Quality Of Service (Qos) pada Jaringan Telekomunikasi Data 4G LTE di Kelurahan Bambu Pemali Kota Merauke," *MUSTEK ANIM HA*, vol. 11, no. 1, pp. 1–6, Apr. 2022, doi: 10.35724/mustek.v11i1.4400.
- [8] Satria Turangga, Martanto, and Yudhistira Arie Wijaya, "Analisis Internet Menggunakan Parameter Quality Of Service pada Alfamart TUPAREV 70," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 1, pp. 392–398, Apr. 2022, doi: 10.36040/jati.v6i1.4693.
- [9] A. Wantoro, "Sistem Monitoring Perawatan dan Perbaikan Fasilitas Gardu PT PLN Area Kota Metro," *Jurnal Tekno Kompak*, vol. 15, no. 1, p. 116, Feb. 2021, doi: 10.33365/jtk.v15i1.903.
- [10] T. M. Raihan, "Sistem Pemantauan Kualitas Air Menggunakan ESP32 dengan Fuzzy Logic Sugeno Berbasis Arduino," 2022.
- [11] "Prototipe Sistem Pemantauan dan Pengendalian Beban Listrik Berbasis Internet Of Things (IoT) dengan Aplikasi Cayenne." <https://ejournal.akprind.ac.id/index.php/jurtek/article/view/1392/108>

- [12] I. Ferdiansyah, "Internet of Things (IoT) : Pengertian dan Manfaat di Berbagai Aspek Bidang Kehidupan," *Liputan Kendal Terkini*, Jun. 28, 2022. Accessed: May 31, 2023. [Online]. Available: <https://www.liputankendalterkini.com/kendal/pr-3613766700/internet-of-things-iot-pengertian-dan-manfaat-di-berbagai-aspek-bidang-kehidupan>
- [13] M. Zaini, S. Safrudin, and Moh. Bachrudin, "Perancangan Sistem Monitoring Tegangan, Arus Dan Frekuensi Pada Pembangkit Listrik Tenaga Mikrohidro Berbasis IoT," *TESLA: Jurnal Teknik Elektro*, vol. 22, no. 2, p. 139, Nov. 2020, doi: 10.24912/tesla.v0i0.9081.
- [14] "Ubidots — Powerful but simple Industrial IoT." <https://ubidots.com/> (accessed Nov. 01, 2023).
- [15] M. I. Q, "Platform IoT," Telkom University, May 25, 2022. <https://miqbal.staff.telkomuniversity.ac.id/platform-iot/> (accessed May 25, 2023).
- [16] I. Update, "Tutorial Thingspeak Lengkap : Pengiriman dan Penerimaan Data," *Indobot Academy - Artikel Baru Seputar Teknologi*, Feb. 19, 2022. Accessed: May 31, 2023. [Online]. Available: <https://indobot.co.id/blog/tutorial-thingspeak-lengkap-pengiriman-dan-penerimaan-data/>
- [17] T. F. Silallagan, M. Firmansyah, and Jaja, "Machine Learning Alat Monitoring Fermentasi Tape Berbasis IoT Menggunakan Metode Algoritma C4.5 pada Platform Thing Speak," *Jurnal Ilmiah Fakultas Ilmu Komputer Universitas Subang*, vol. 10, no. 1, Jan. 2023.
- [18] A. Hadinata and Mashoedah, "Internet of Things-based Hydroponic: Literature Review - IOPscience," *Journal of Physics: Conference Series*, vol. 2111, no. 1, Nov. 2021, doi: 10.1088/1742-6596/2111/1/012014.
- [19] E. B. Wagiu, A. Butar-butur, and J. I. Sihotang, "Analisis QoS (Quality of Service) pada Jaringan Internet (Studi Kasus: Universitas Advent Indonesia)," *TelKa*, vol. 9, no. 01, pp. 31–41, Apr. 2019, doi: 10.36342/teika.v9i01.789.
- [20] C. T. Helena Manurung, J. Arifin, F. T. Syifa, and R. A. Rochmanto, "Pemanfaatan ESP32 Sebagai Sistem Pemantauan Kualitas Air Keran Siap Minum Secara Real-Time Menggunakan Aplikasi," *Journal of Telecommunication, Electronics, and Control Engineering (JTECE)*, vol. 4, no. 2, pp. 93–98, Jul. 2022, doi: 10.20895/jtece.v4i2.535.
- [21] T. M. Diansyah, "Analisa Pencegahan Aktivitas Ilegal di Dalam Jaringan Menggunakan Wireshark," *Jurnal TIMES*, vol. 4, no. 2, pp. 20–23, Jan. 2015.
- [22] "Wireshark icon new," *Wikimedia Commons*, May 09, 2023. https://commons.wikimedia.org/wiki/File:Wireshark_icon_new.png

- (accessed Nov. 01, 2023).
- [23] M. I. Q, "Protokol MQTT," Telkom University, Sep. 15, 2021. https://miqbal.staff.telkomuniversity.ac.id/protokol_mqtt/ (accessed May 25, 2023).
 - [24] "MQTT Specification." <https://mqtt.org/mqtt-specification/> (accessed May 31, 2023).
 - [25] C.-W. Lee and D.-G. Yoo, "Development of Leakage Detection Model and Its Application for Water Distribution Networks Using RNN-LSTM," *Sustainability*, vol. 13, no. 16, Aug. 2021, doi: 10.3390/su13169262.
 - [26] D. Kurniasih, S. Mariani, and S. Sugiman, "Efisiensi Relatif Estimator Fungsi Kernel Gaussian Terhadap Estimator Polinomial dalam Peramalan USD Terhadap JPY," *Unnes Journal of Mathematics*, vol. 2, no. 2, Nov. 2013, doi: 10.15294/ujm.v2i2.3245.
 - [27] F. Fansuri, "Sistem Peramalan Dan Monitoring Persediaan Obat di RSPG Cisarua Bogor dengan Menggunakan Metode Single Exponential Smoothing dan Reorder Point." *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*
 - [28] A. Chugh, "MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?," *Analytics Vidhya*, Mar. 16, 2022. Accessed: Nov. 01, 2023. [Online]. Available: <https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e>
 - [29] B. I. Sanny and R. K. Dewi, "Pengaruh Net Interest Margin (NIM) Terhadap Return on Asset (ROA) pada PT Bank Pembangunan Daerah Jawa Barat Dan Banten Tbk Periode 2013- 2017" *Jurnal E-Bis (Ekonomi-Bisnis)*, vol. 4, no. No. 1 (2020), pp. 74–87, Jul. 2020.
 - [30] I. Academy, "Tutorial Arduino UNO R3 Mengakses Water Flow Sensor," *Indobot Academy - Artikel Baru Seputar Teknologi*, May 30, 2022. Accessed: May 31, 2023. [Online]. Available: <https://indobot.co.id/blog/tutorial-arduino-uno-r3-mengakses-water-flow-sensor/>
 - [31] "ESP32," Himpunan Mahasiswa Teknik Komputer. <https://student-activity.binus.ac.id/himtek/2022/07/27/esp32/> (accessed May 25, 2023).
 - [32] Erintafifah, "Mengenal Perangkat Lunak Arduino IDE," *KMTek*, Oct. 07, 2021. <https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide> (accessed May 25, 2023).
 - [33] R. Rahmatina, M. N. Aripin, M. Ikbal, and A. Deolika, "Implementasi Transistor BD139 dan Rangkaian Relay pada Mesin Air," *Journal of Information Technology*, vol. 3, no. 1, pp. 11–18, Mar. 2023, doi: 10.46229/jifotech.v3i1.579.

- [34] L. Khakim, I. Afriliana, N. Nurohim, and A. Rakhman, "Proteksi Kebocoran Gas LPG Rumah Tangga Berbasis Mikrokontroler," *Komputika : Jurnal Sistem Komputer*, vol. 11, no. 1, pp. 40–47, Jan. 2022, doi: 10.34010/komputika.v11i1.4977.
- [35] Y. Triafandy, A. B. Pulungan, and H. Hamdani, "Kendali Solar Tracker Menggunakan Selenoid Valve sebagai Pengendali Aliran fluida," *JTEIN: Jurnal Teknik Elektro Indonesia*, vol. 1, no. 2, pp. 174–178, Nov. 2020, doi: 10.24036/jtein.v1i2.66.
- [36] P. R. Utami, "Analisis Perbandingan Quality Of Service Jaringan Internet Berbasis Wireless pada Layanan Internet Service Provider (ISP) Indihome dan First Media," *Jurnal Ilmiah Teknologi dan Rekayasa*, vol. 25, no. 2, pp. 125–137, Oct. 2020, doi: 10.35760/tr.2020.v25i2.2723.
- [37] H. Kusbandono and E. M. Syafitri, "Penerapan Quality Of Service (QoS) dengan Metode PCQ untuk Manajemen Bandwidth Internet pada WLAN Politeknik Negeri Madiun," *RESEARCH: Journal of Computer, Information System & Technology Management*, vol. 2, no. 1, pp. 7–12, Apr. 2019.
- [38] J. Badriyah, A. Fariza, and T. Harsono, "Prediksi Curah Hujan Menggunakan Long Short Term Memory," *JURNAL MEDIA INFORMATIKA BUDIDARMA*, vol. 6, no. 3, pp. 1297–1303, Jul. 2022.
- [39] K. A. Pranoto *et al.*, "Comparison Analysis of Data Sending Performance Using The Cayenne and ThingSpeak IoT Platform," in *2022 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*, Nov. 2022. Accessed: Oct. 10, 2023. [Online]. Available: <http://dx.doi.org/10.1109/icimcis56303.2022.10017648>
- [40] "Plans & Billing: What is the difference between Ubidots and Ubidots STEM?," Ubidots Help Center. <https://help.ubidots.com/en/articles/639806-plans-billing-what-is-the-difference-between-ubidots-and-ubidots-stem> (accessed Jan. 04, 2024).
- [41] "Licensing FAQ," ThingSpeak IoT. https://thingspeak.com/pages/license_faq (accessed Jan. 04, 2024).
- [42] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, Mar. 2019, doi: 10.1016/j.patrec.2018.02.010.

Biodata



Nama : Asri Arganing Hasanah
TTL : Dumai, 5 April 2002
Agama : Islam
Alamat : Jl. Cemara Angin No. 8, Bumi Ayu, Kota Dumai
Email : asriarganing@gmail.com
Riwayat Pendidikan SMA/SMK : SMA Negeri 1 Kota Dumai
SMP : SMP Negeri Binaan Khusus Kota Dumai
SMP Negeri 12 Kota Batam



Nama : Nurul Khairani
TTL : Payakumbuh, 12 Juli 2001
Agama : Islam
Alamat : Bengkong Indah 3 Blok B No. 06, Kota Batam
Email : khairani.nurul1701@gmail.com
Riwayat Pendidikan :
SMA/SMK : SMA Negeri 8 Kota Batam
SMP : SMP Negeri 30 Kota Batam

Lampiran

A. Aplikasi



Gambar 48. Ubidots Pengujian 1



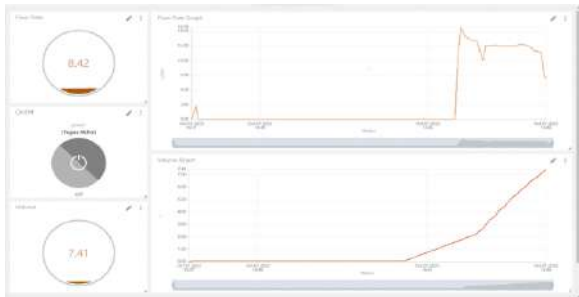
Gambar 49. ThingSpeak Pengujian 1



Gambar 50. *Ubidots* Pengujian 2



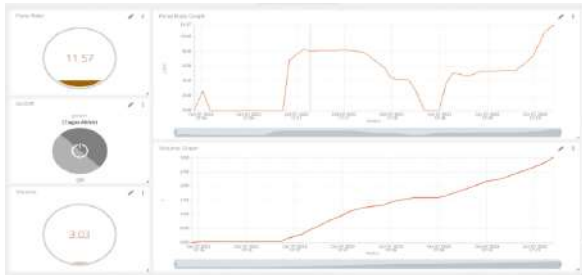
Gambar 51. *ThingSpeak* Pengujian 2



Gambar 52. *Ubidots* Pengujian 3



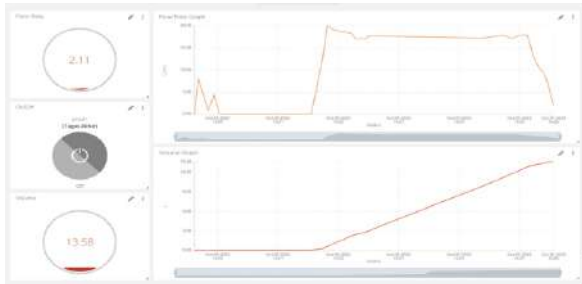
Gambar 53. *ThingSpeak* Pengujian 3



Gambar 54. Ubidots Pengujian 4



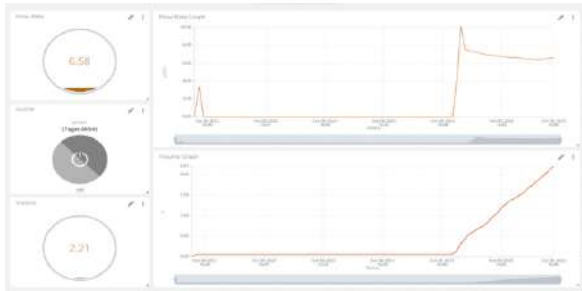
Gambar 55. ThingSpeak Pengujian 4



Gambar 56. Ubidots Pengujian 5



Gambar 57. ThingSpeak Pengujian 5



Gambar 58. Ubidots Pengujian 6



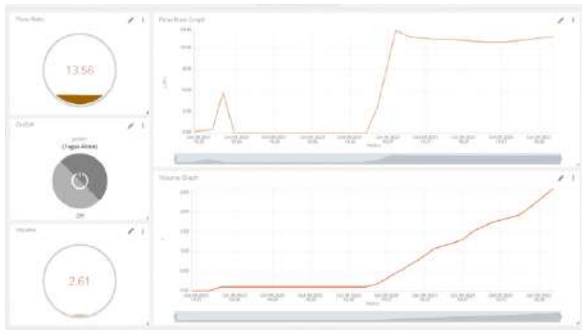
Gambar 59. ThingSpeak Pengujian 6



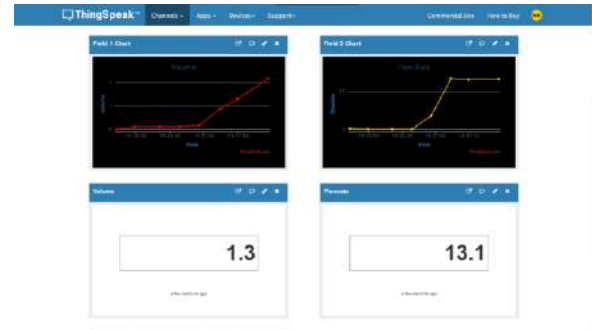
Gambar 60. Ubidots Pengujian 7



Gambar 61. ThingSpeak Pengujian 7



Gambar 62. Ubidots Pengujian 8



Gambar 63. ThingSpeak Pengujian 9

B. Meteran Manual



Gambar 64. Meteran Manual Sebelum Pengujian 1



Gambar 65. Meteran Manual Sesudah Pengujian 1



Gambar 66. Meteran Manual Sebelum Pengujian 2



Gambar 67. Meteran Manual Sesudah Pengujian 2



Gambar 68. Meteran Manual Sebelum Pengujian 3



Gambar 69. Meteran Manual Sesudah Pengujian 3



Gambar 70. Meteran Manual Sebelum Pengujian 4



Gambar 71. Meteran Manual Sesudah Pengujian 4



Gambar 72. Meteran Manual Sebelum Pengujian 5



Gambar 73. Meteran Manual Sesudah Pengujian 5



Gambar 74. Meteran Manual Sebelum Pengujian 6



Gambar 75. Meteran Manual Sesudah Pengujian 6



Gambar 76. Meteran Manual Sebelum Pengujian 7



Gambar 77. Meteran Manual Sesudah Pengujian 7



Gambar 78. Meteran Manual Sebelum Pengujian 8



Gambar 79. Meteran Manual Sesudah Pengujian 8

C. Statistik QoS

Statistics

Measurement	Captured	Displayed
Packets	2188	401 (18.3%)
Time span, s	351.329	350.773
Average pps	6.2	1.1
Average packet size, B	183	76
Bytes	400573	30415 (7.6%)
Average bytes/s	1140	86
Average bits/s	9121	693

Gambar 80. *Ubidots* Pengujian 1

Statistics

Measurement	Captured	Displayed
Packets	4536	76 (1.7%)
Time span, s	348.246	314.716
Average pps	13.0	0.2
Average packet size, B	367	152
Bytes	1663464	11553 (0.7%)
Average bytes/s	4776	36
Average bits/s	38 k	293

Gambar 81. *ThingSpeak* Pengujian 1

Statistics

Measurement	Captured	Displayed
Packets	1996	329 (16.5%)
Time span, s	309.063	303.917
Average pps	6.5	1.1
Average packet size, B	147	77
Bytes	293693	25226 (8.6%)
Average bytes/s	950	83
Average bits/s	7602	664

Gambar 82. *Ubidots* Pengujian 2

Statistics

Measurement	Captured	Displayed
Packets	10512	103 (1.0%)
Time span, s	326.187	208.168
Average pps	32.2	0.5
Average packet size, B	639	196
Bytes	6715297	20163 (0.3%)
Average bytes/s	20 k	96
Average bits/s	164 k	774

Gambar 83. *ThingSpeak* Pengujian 2

Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	2150	343 (16.0%)
Time span, s	323.996	318.582
Average pps	6.6	1.1
Average packet size, B	143	76
Bytes	307473	26025 (8.5%)
Average bytes/s	949	81
Average bits/s	7592	653

Gambar 84. Ubidots Pengujian 3**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	2506	350 (14.0%)
Time span, s	313.263	310.161
Average pps	8.0	1.1
Average packet size, B	150	82
Bytes	376549	28815 (7.7%)
Average bytes/s	1202	92
Average bits/s	9616	743

Gambar 86. Ubidots Pengujian 4**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	1541	317 (20.6%)
Time span, s	307.426	299.450
Average pps	5.0	1.1
Average packet size, B	127	84
Bytes	196404	26580 (13.5%)
Average bytes/s	638	88
Average bits/s	5110	710

Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	3952	53 (1.3%)
Time span, s	319.659	79.743
Average pps	12.4	0.7
Average packet size, B	339	139
Bytes	1338099	7380 (0.6%)
Average bytes/s	4186	92
Average bits/s	33 k	740

Gambar 85. ThingSpeak Pengujian 3**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	3274	234 (7.1%)
Time span, s	316.664	310.459
Average pps	10.3	0.8
Average packet size, B	309	157
Bytes	1010835	36680 (3.6%)
Average bytes/s	3192	118
Average bits/s	25 k	945

Gambar 87. ThingSpeak Pengujian 4**Statistics**

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	3077	202 (6.6%)
Time span, s	305.907	302.060
Average pps	10.1	0.7
Average packet size, B	361	140
Bytes	1110534	28198 (2.5%)
Average bytes/s	3630	93
Average bits/s	29 k	746

Gambar 88. Ubidots Pengujian 5

Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	2711	323 (11.9%)
Time span, s	309.624	303.211
Average pps	8.8	1.1
Average packet size, B	180	77
Bytes	488038	25004 (5.1%)
Average bytes/s	1576	82
Average bits/s	12 k	659

Gambar 90. Ubidots Pengujian 6

Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	2488	361 (14.5%)
Time span, s	323.005	322.353
Average pps	7.7	1.1
Average packet size, B	154	84
Bytes	382084	30182 (7.9%)
Average bytes/s	1182	93
Average bits/s	9463	749

Gambar 92. Ubidots Pengujian 7

Gambar 89. ThingSpeak Pengujian 5

Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	3168	183 (5.8%)
Time span, s	307.012	276.732
Average pps	10.3	0.7
Average packet size, B	302	140
Bytes	956986	25691 (2.7%)
Average bytes/s	3117	92
Average bits/s	24 k	742

Gambar 91. ThingSpeak Pengujian 6

Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	3361	166 (4.9%)
Time span, s	301.635	287.668
Average pps	11.1	0.6
Average packet size, B	296	147
Bytes	993860	24483 (2.5%)
Average bytes/s	3294	85
Average bits/s	26 k	680

Gambar 93. ThingSpeak Pengujian 7

Statistics

<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	1535	264 (17.2%)
Time span, s	302.994	295.588
Average pps	5.1	0,9
Average packet size, B	123	81
Bytes	189547	21349 (11.3%)
Average bytes/s	625	72
Average bits/s	5004	577

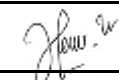



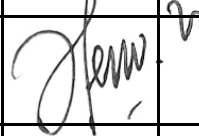





Gambar 94. Ubidots Pengujian 8**Statistics**





<u>Measurement</u>	<u>Captured</u>	<u>Displayed</u>
Packets	3540	144 (4.1%)
Time span, s	302.926	267.025
Average pps	11.7	0.5
Average packet size, B	310	161
Bytes	1097034	23183 (2.1%)
Average bytes/s	3621	86
Average bits/s	28 k	694

Gambar 95. ThingSpeak Pengujian 8

FORMULIR LOGBOOK BIMBINGAN DAN PENGAJUAN SIDANG TUGAS AKHIR

Nama : Nurul Khairani
 NIM : 4212001044
 Pembimbing I : Heru Wijanarko, S.T., M. Sc
 Pembimbing II : Muhammad Naufal Airlangga, S.Pd., M.P.H
 Judul : Analisa Performansi Pengiriman Data Menggunakan *Platform* IoT *Ubidots* dan *Thingspeak*, Studi Kasus: Monitoring Aliran Air Rumah Pintar

No	Hari/Tanggal	Rincian Kegiatan	TTD Pembimbing	
			I	II
1.	Sabtu, 12 Agustus 2023	Laporan progres tugas akhir		
2.	Selasa, 15 Agustus 2023	Diskusi mengenai Bab 1, Bab 2, Bab 3		
3.	Kamis, 31 Agustus 2023	Diskusi mengenai kalibrasi sensor, pengujian sensor dan problem website		
4.	Selasa, 12 September 2023	Diskusi mengenai progress yang telah di kerjakan		
5.	Kamis, 14 September 2023	Diskusi mengenai pergantian website cayenne		
6.	Rabu, 20 September 2023	Diskusi terkait perubahan website cayenne dan progress yang telah dikerjakan		
7.	Rabu, 27 September 2023	Diskusi progress yang telah dikerjakan dan problem yang dihadapi		
8.	Rabu, 18 Oktober 2023	Diskusi progress yang telah dikerjakan dan problem yang dihadapi		
9.	Jumat, 3 November 2023	Diskusi mengenai problem yang dihadapi terkait RNN-LSTM		
10.	Selasa, 7 November 2023	Diskusi mengenai saran dan masukan terhadap problem yang dihadapi terkait RNN-LSTM		

11.	Selasa, 28 November 2023	Diskusi mengenai penulisan tugas akhir		
12.	Kamis, 7 Desember 2023	Finalisasi tugas akhir dan persetujuan maju sidang		
13.	Senin, 11 Desember 2023	Finalisasi tugas akhir dan persetujuan maju sidang		
14.	Selasa, 12 Desember 2023	Finalisasi tugas akhir dan persetujuan maju sidang		

Berdasarkan hasil bimbingan yang telah dilaksanakan selama 5 bulan dan telah disetujui oleh dosen pembimbing, maka dengan ini saya mengajukan diri sebagai peserta Sidang Tugas Akhir

Batam, 12 Desember 2023



Nurul Khairani
NIM : 4212001044