

Implementasi *Single-Hop Wireless Sensor Network* pada Lintasan Mobil Remote Control

LISA KRISTIANA, MUHAMMAD RAFLY

Program Studi Informatika, Institut Teknologi Nasional Bandung
Email : lisa@itenas.ac.id

Received 15 Mei 2022 | *Revised* 2 Juni 2022 | *Accepted* 24 Juni 2022

ABSTRAK

Penerapan teknologi dalam sistem pemantauan dapat mengatasi kelemahan pada pemantauan secara manual. Untuk meningkatkan kualitas pemantauan, diperlukan suatu sistem yang dapat memantau mobil remote control yang berada di lintasan. Wireless Sensor Network (WSN) merupakan teknologi yang dapat meningkatkan efisiensi pemantauan secara otomatis pada area tertentu. WSN pada lintasan menerapkan sistem komunikasi Single-Hop. Sistem Single-Hop menggunakan sensor node (obstacle infrared) untuk mendeteksi mobil di area lintasan, lalu dikirim langsung oleh sink node ke server dengan menggunakan ESP8266-01. Arsitektur WSN menggunakan Topologi Point to Point terhubung langsung antara sensor node dengan Access Point. Metode pengiriman Single-Hop menggunakan Transmission Control Protocol/Internet Protocol (TCP/IP). Hasil pengujian TCP/IP berdasarkan jarak baca sensor menggunakan parameter Quality of Service (QoS) meliputi parameter Delay, Jitter, Throughput, Packet loss, menghasilkan delay, jitter dan throughput yang sangat baik dan stabil, dan paket loss 0%. Kemudian TCP/IP menguji tingkat akurasi pengiriman data ke server sebanyak 400 kali menghasilkan tingkat akurasi 100%.

Kata kunci: *lintasan mobil remote control, single-hop wireless sensor network*

ABSTRACT

Application of technology in monitoring system can overcome weaknesses in manual monitoring. To improve quality of monitoring, need system that can monitor remote control cars on track. Wireless Sensor Network (WSN) is technology that can increase monitoring efficiency automatically in certain areas. WSN on track implements Single-Hop communication system. Single-Hop system uses sensor node (obstacle infrared) to detect cars in the track area, then sent directly by sink node to server using ESP8266-01. WSN architecture uses Point to Point topology connected directly between sensor nodes and Access Point. Single-Hop delivery method uses Transmission Control Protocol/Internet Protocol (TCP/IP). TCP/IP test results based on sensor reading distance using Quality of Service (QoS) parameters including Delay, Jitter, Throughput, Packet loss parameters, resulting in excellent and stable delay, jitter and throughput, and 0% packet loss. Then TCP/IP tests the accuracy of sending data to server 400 times resulting in an accuracy rate of 100%.

Keywords: *remote control car track, single-hop wireless sensor network*

1. PENDAHULUAN

Pada dasarnya *Wireless Sensor Network* adalah jaringan yang terdiri dari sensor node dan node-node tersebut bekerja sama dapat melakukan pemantauan area, situasi atau lingkungan tertentu lalu node tersebut memiliki kemampuan penginderaan atau pengukuran besaran fisis, memproses, menyimpan serta memiliki kemampuan komunikasi yang dibangun tanpa infrastruktur yang tetap **(Hulu, Riyanto T, & Widyanoro, 2015)**. Pada studi kasus lintasan mobil *remote control* yang ditemukan, terdapat beberapa sensor yang berada dalam satu lingkungan pada lintasan mobil *remote control* yang berfungsi untuk deteksi mobil *remote control* apabila mendekati area tertentu, namun pemantauan tersebut saat ini masih dilakukan secara manual yaitu dengan mengandalkan tenaga manusia untuk pencatatan mobil *remote control* yang melewati batas area yang telah ditentukan.

Pencatatan tersebut dinilai kurang efisien, sehingga membutuhkan banyak waktu dan tenaga dalam mengaplikasikannya. Penerapan teknologi WSN dapat menjadi solusi pada sistem pendeteksian pada area lingkungan tertentu yang dapat memantau kondisi suhu, kelembaban, gas dan curah hujan secara *realtime* sehingga memudahkan pihak yang berkompeten terkait pengendalian pencemaran pada area lingkungan tersebut serta dapat dijadikan sebagai bahan kajian **(Nurkamid & Widodo, 2021)**. Maka dari itu pada penelitian ini mengimplementasikan WSN pada lintasan mobil *remote control*.

WSN merupakan perangkat jaringan yang saling terhubung yang terdiri dari *sensor node*, *sink node*, dan *router*. Perangkat tersebut saling terhubung satu sama lain dan dapat berkomunikasi satu sama lain secara langsung tanpa perlu infrastruktur jaringan seperti *Access Point* atau *router* **(Nashrullah, Primananda, & Widasari, 2018)**. *Sink node* berfungsi mengumpulkan informasi yang diperoleh dari *sensor node* yang berada di lintasan mobil *remote control*, sehingga tidak memerlukan tenaga manusia lagi untuk memantau keadaan mobil *remote control* yang melewati batas area yang sudah ditentukan.

Sistem *Multi Sensor Network* (MSN) merupakan teknologi *Wireless Sensor Network* yang memanfaatkan lebih dari satu sensor dalam satu perangkat. Sensor ini memiliki kemampuan seperti sinyal untuk dapat mentransmisikan data sensor yang dihasilkan ke pusat pemantauan secara otomatis dan hasil data tentang objek yang disensor dilakukan secara *real-time* **(Tarmidi, Taqwa, & Handayani, 2019)**. Pada sistem lintasan mobil *remote control* yang akan dibuat menggunakan modul *Wi-Fi* ESP8266-01 sebagai media komunikasi dengan *server*, sehingga dapat melakukan pengiriman data yang dihasilkan oleh sensor secara *real-time*.

Implementasi WSN dapat diterapkan dengan *Single-Hop* agar *router* dapat menerima data sensor lebih dari satu pengirim. Metode *Single-Hop* sangat memungkinkan pengiriman data sensor lebih cepat dikarenakan masing-masing node pengirim langsung terhubung dengan *node* pusat atau *router* **(Subono, Hidayat, & Akhmad, 2019)**. Pada penelitian ini akan mengimplementasikan *Wireless Sensor Network* pada lintasan mobil *remote control* dalam melakukan pemantauan pada area lintasan mobil *remote control* serta menerapkan *Single-Hop* pada proses pengiriman data ke *server* sehingga *node sensor* tersebut langsung dikirim menuju *node* pusat agar meminimalisir terjadinya *packet loss* dan *delay* pada proses pengiriman data. Jadi ketika melakukan proses pengiriman data tidak perlu lagi mengirim dari *node sensor* satu ke *node sensor* yang lainnya dan diharapkan pada proses pengiriman data tersebut dapat menjadi lebih efisien dan hemat biaya.

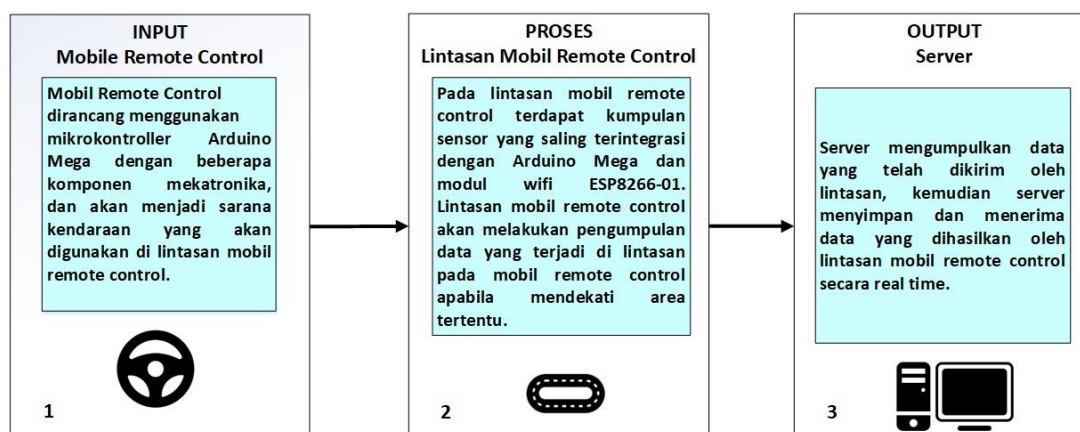
Berdasarkan hasil penelitian di atas maka pada penelitian ini merancang dan mengimplementasikan *Single-Hop Wireless Sensor Network* pada lintasan mobil *remote control*. Sistem dirancang memanfaatkan komponen utama seperti Arduino Mega 2560 sebagai *control system*, sensor *obstacle avoidance* sebagai sensor *node* yang digunakan untuk membaca mobil *remote control* yang ada di lintasan mobil remote control, ESP8266-01 sebagai modul wifi agar sistem dapat terhubung dengan server dan mengirim data-data yang dihasilkan oleh sensor ke server secara *real-time*.

2. METODE PENELITIAN

Pada penelitian ini metode yang digunakan adalah *prototype*. Pada bagian ini meliputi blok diagram, *work flow*, arsitektur jaringan, dan *flowchart* sistem.

2.1 Perancangan Sistem

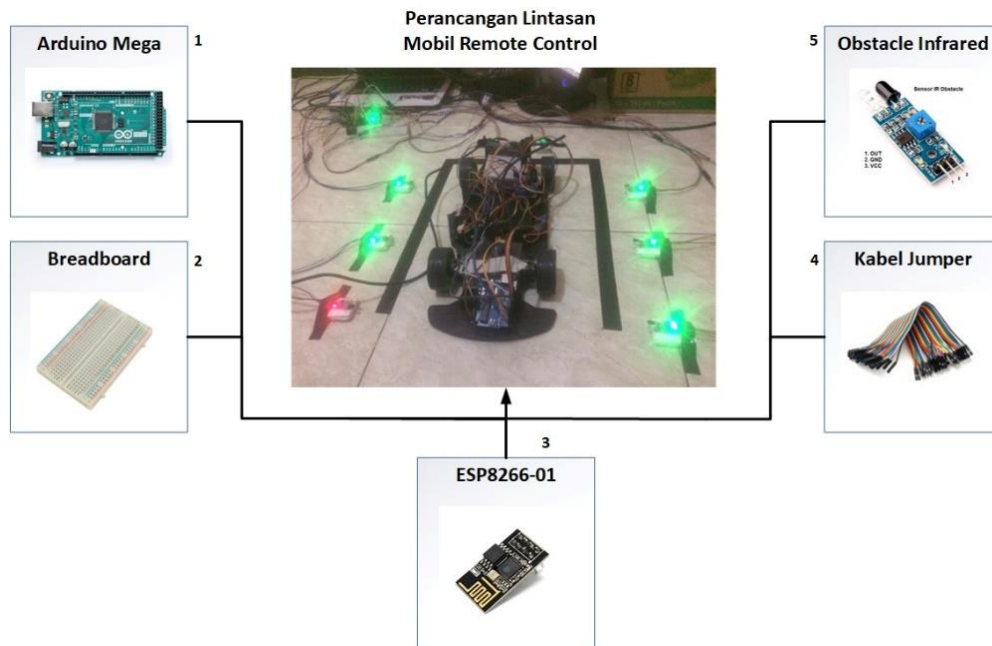
Pada Gambar 1 merupakan perancangan keseluruhan sistem. Sistem ini memiliki 3 (tiga) komponen utama, yaitu (1) Mobil *remote control* dengan berbagai macam mekatronika di dalamnya yang menjadi sebagai sarana kendaraan yang digunakan pada lintasan mobil *remote control*. (2) Lintasan mobil *remote control* yang terintegrasi dengan banyak mekatronika dan akan memantau mobil *remote control* yang berada di lintasan mobil remote control. (3) Server berfungsi sebagai penerima dan penyimpanan data yang dihasilkan oleh lintasan mobil *remote control* ketika sedang memantau mobil *remote control* di lintasan.



Gambar 1. Blok Diagram Keseluruhan

2.2 Perancangan Sistem Lintasan Mobil *Remote Control*

Pada Gambar 2 merupakan blok diagram pada perancangan *implementasi Single-Hop Wireless Sensor Network* pada lintasan mobil *remote control*. Tahapan selanjutnya dijelaskan bahwa skema perangkat terhubung pada sistem yang telah dibuat berdasarkan pembatas yang telah dibuat sebagai berikut :

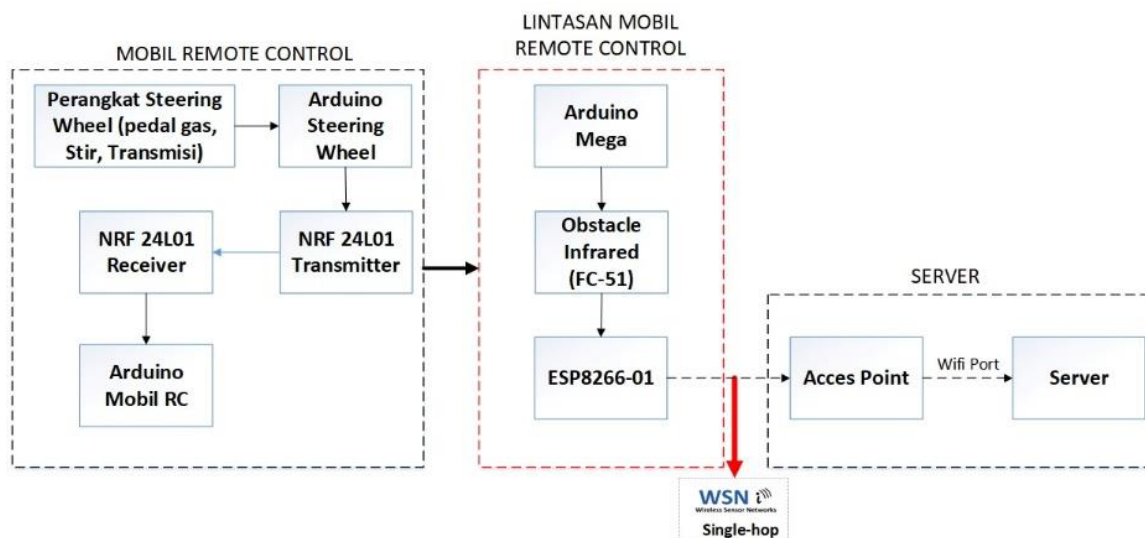


Gambar 2. Blok Diagram Lintasan Mobil Remote Control

1. Arduino Mega digunakan sebagai *sink node* dan *control system* yang memproses *input* dan *output* yang terjadi pada lintasan mobil *remote control*.
2. Breadboard digunakan sebagai penghubung kabel jumper dengan Arduino mega dan sensor yang digunakan untuk membuat rangkaian elektronik sementara untuk tujuan uji coba atau *prototype*.
3. ESP8266-01 digunakan sebagai modul *Wi-Fi* yang berfungsi sebagai penghubung Arduino Mega agar dapat terhubung langsung dengan *Wi-Fi* dan membuat koneksi TCP/IP.
4. Kabel *jumper* digunakan sebagai penghubung dengan *Breadboard*, menghubungkan antar titik pada pcb *single slide* dan juga dapat digunakan untuk menghubungkan jalur rangkaian yang terputus dengan cara menjumpernya.
5. *Obstacle Infrared* digunakan sebagai *sensor node* yang digunakan yang ada pada lintasan mobil *remote control* yang berfungsi untuk mendeteksi mobil *remote control* di lintasan apabila melewati pembatas jalan yang telah ditentukan.

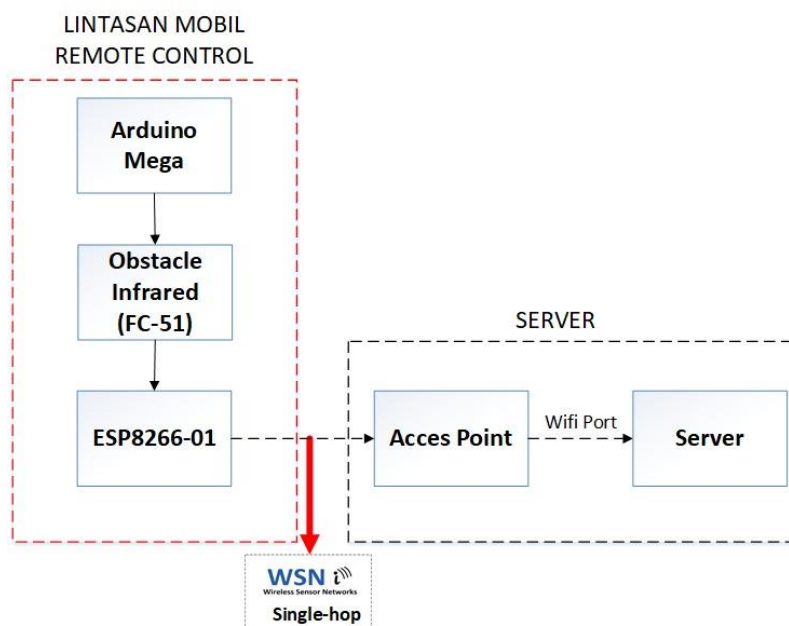
2.3 Workflow

Pada Gambar 3 merupakan cara kerja sistem keseluruhan, prosesnya dijelaskan sebagai berikut, mobil *remote control* sebagai objek yang dideteksi di lintasan mobil *remote control* kemudian data tersebut dikirim dan disimpan di *server* sebagai hasil pemantauan.



Gambar 3. Workflow Sistem Keseluruhan

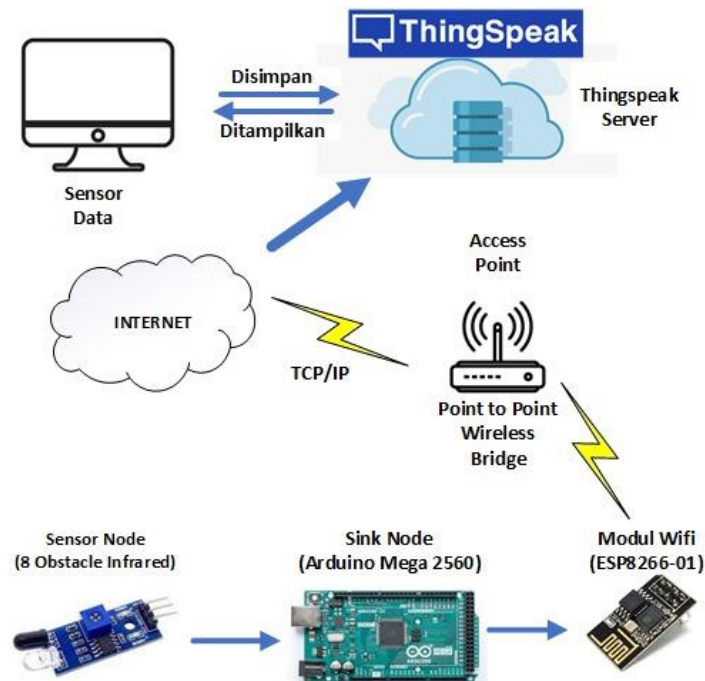
Pada penelitian ini, sistem yang dirancang hanya berfokus pada implementasi *Single-Hop* WSN pada lintasan mobil *remote control* (Zhao, Qiao, Sudhaakar, & Yoon, 2013). Pada lintasan mobil *remote control* terdapat Arduino Mega sebagai *control system* yang terhubung dengan *obstacle infrared* sebagai *sensor node* untuk memantau mobil *remote control* yang berada di lintasan. kemudian Arduino Mega juga terhubung dengan modul *Wi-Fi* yaitu ESP8266-01 yang berfungsi sebagai *sink node* untuk mengumpulkan data yang dihasilkan oleh *sensor node* yang berada di lintasan dan juga sebagai penghubung ke *access point* agar lintasan mobil *remote control* dapat terhubung dengan *server* dan dapat mengirim data yang dihasilkan di lintasan secara *real-time*. Selanjutnya menguji proses pengiriman data yang dihasilkan oleh sensor yang ada di lintasan mobil *remote control* yang selanjutnya data yang dihasilkan tersebut dikirim ke *server* dan disimpan untuk diolah datanya lebih lanjut, berikut Gambar 4 adalah *workflow* yang berfokus pada sistem lintasan mobil *remote control* dengan server :



Gambar 4. Workflow Single-Hop WSN pada Lintasan Mobil Remote Control

2.4 Arsitektur Jaringan

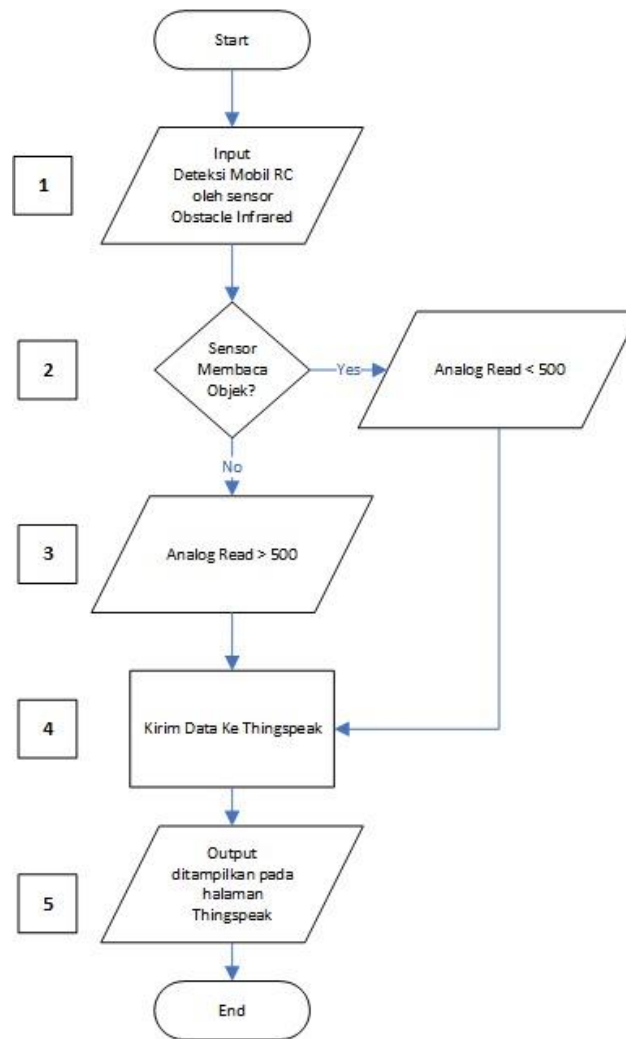
Gambar 5 merupakan arsitektur jaringan pada lintasan mobil *remote control*, dapat dijelaskan bahwa terdapat 8 buah *sensor node* yang digunakan pada lintasan mobil *remote control* dan terhubung dengan Arduino Mega 2560 yang berfungsi sebagai *sink node*. Agar dapat terhubung ke *access point* dan melakukan pengiriman data ke *server*, Arduino Mega memerlukan modul *Wi-Fi* tambahan yaitu ESP8266-01. Arduino Mega yang sudah terhubung dengan ESP8266-01 selanjutnya Arduino Mega melakukan koneksi dengan *Access Point* secara *Point to Point*. Kemudian *sensor node* yang dihasilkan oleh sensor *obstacle infrared* pada lintasan mobil *remote control* dapat dikirim ke *server* menggunakan protokol *Transmission Control Protocol/Internet Protocol* (TCP/IP) melalui internet (**Hidayati & Suwadi, 2016**), kemudian agar dapat terhubung dengan *web server Thingspeak* diperlukan *Application Programming Interface* (API) pada sistem yang dibuat sehingga data sensor yang dikirim dapat tersimpan dan dilihat melalui web server *Thingspeak*. Kemudian hasil sensor yang disimpan di server *Thingspeak* dapat dilihat melalui laptop/*handphone* secara *real time*.



Gambar 5. Arsitektur Sistem Jaringan

2.5 Flowchart sistem

Gambar 6 merupakan *flowchart* sistem lintasan mobil *remote control*, dapat dijelaskan bahwa pada tahapan awal *sensor node* memasukan deteksi yang dihasilkan oleh mobil *remote control* ketika berada di lintasan. Apabila sensor membaca objek yang ada di depannya, maka sensor akan menghasilkan nilai analog kurang dari 500. Selanjutnya apabila sensor tidak membaca objek yang ada di depannya, maka sensor akan menghasilkan nilai analog lebih dari 500. Setelah itu mengirim data yang dihasilkan oleh sensor node dari ESP8266-01 ke *server Thingspeak*. Kemudian pada tahapan terakhir menampilkan data sensor *node* yang telah dihasilkan di web server *Thingspeak*.



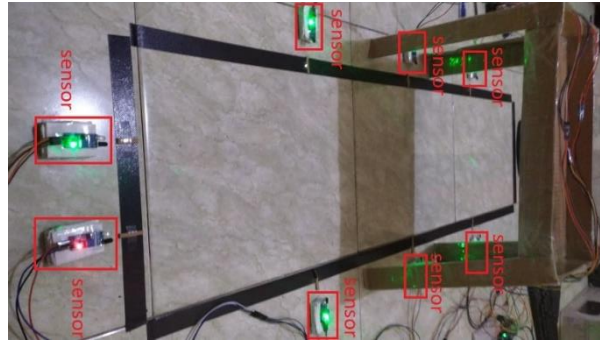
Gambar 5. Flowchart Umum Sistem

3. HASIL DAN PEMBAHASAN

Pada bagian ini terdapat lintasan mobil remote control dibagi menjadi 4 bagian yaitu jalan lurus, zig-zag, parkir *serie*, dan parkir *parallel*. Hasil penelitian meliputi implementasi sensor *node*, *sink node*, dan *Single-Hop WSN* pada pengiriman data ke web server.

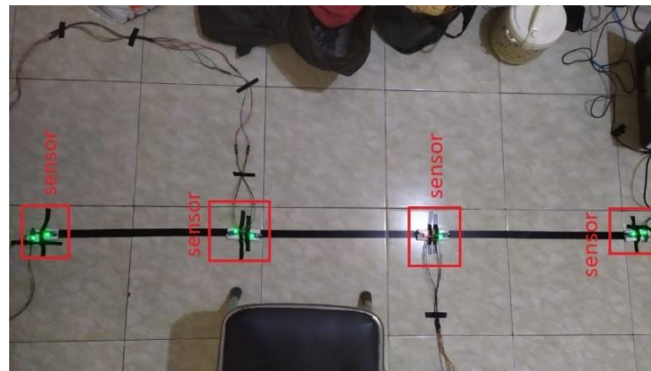
3.1 Implementasi Sensor Node

Implementasi sensor node terdiri dari Sensor *Obstacle Infrared*, dan kabel *jumper* yang terhubung langsung dengan Arduino Mega 2560. Sensor *Obstacle Infrared* berfungsi untuk mendeteksi mobil *remote control* yang sedang berada di lintasan, kemudian data sensor tersebut dikirim langsung melalui *aces point*. Implementasi *sensor node* pada lintasan mobil *remote control* dapat dilihat pada Gambar 6, Gambar 7, Gambar 8, dan Gambar 9.



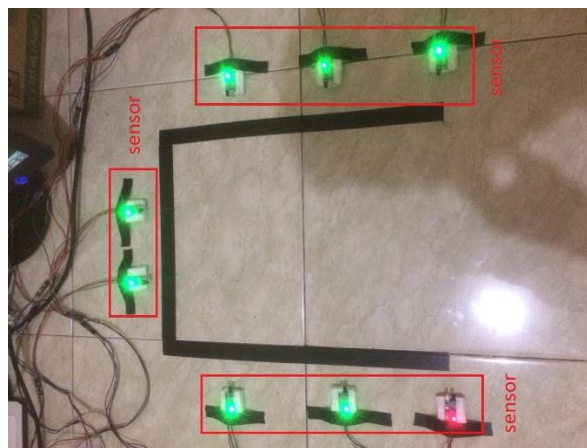
Gambar 6. Implementasi *Sensor Node* Lintasan Lurus

Implementasi *Single-Hop* WSN pada lintasan mobil *remote control* pada Gambar 6 dirangkai sampai berbentuk lintasan lurus kemudian sensor *obstacle avoidance* yang ditandai dengan kotak merah yang berfungsi untuk mendeteksi mobil *remote control* yang keluar jalur lintasan ketika sedang berada di lintasan mobil *remote control*.



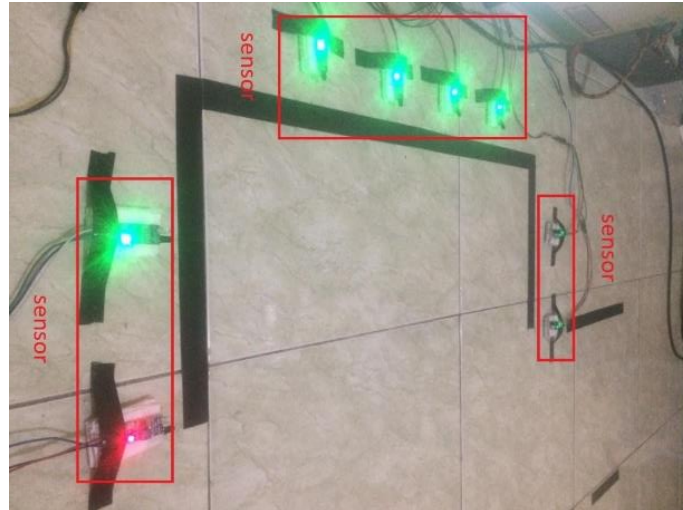
Gambar 7. Implementasi *Sensor Node* Lintasan Zig-Zag

Implementasi *Single-Hop* WSN pada lintasan mobil *remote control* pada Gambar 7 dirangkai sampai berbentuk lintasan zig-zag, kemudian sensor *obstacle avoidance* yang ditandai dengan kotak merah yang berfungsi untuk mendeteksi mobil *remote control* yang keluar jalur lintasan ketika sedang berada di lintasan mobil *remote control*.



Gambar 8. Implementasi *Sensor Node* Parkir *Serie*

Implementasi *Single-Hop* WSN pada lintasan mobil *remote control* pada Gambar 6 dirangkai sampai berbentuk parkir *serie*, kemudian sensor *obstacle avoidance* yang ditandai dengan kotak merah yang berfungsi untuk mendeteksi mobil *remote control* yang keluar jalur lintasan ketika sedang berada di lintasan mobil *remote control*.

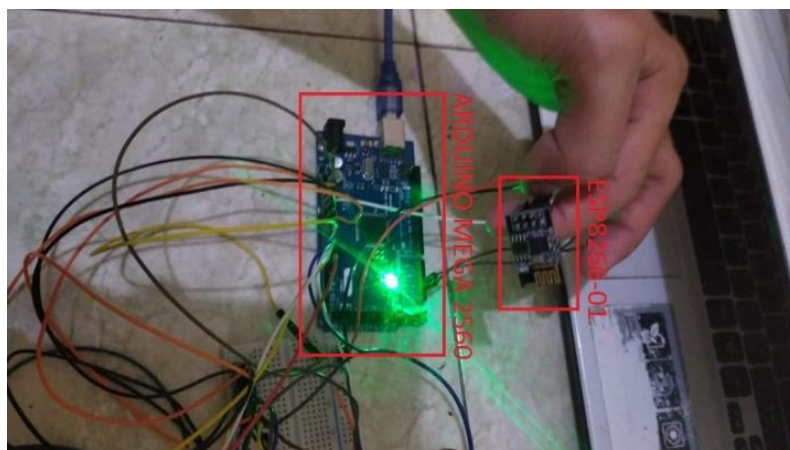


Gambar 9. Implementasi *Sensor Node* Parkir Paralel

Implementasi *Single-Hop* WSN pada lintasan mobil *remote control* pada Gambar 9 dirangkai sampai berbentuk parkir paralel, kemudian sensor *obstacle avoidance* yang ditandai dengan kotak merah yang berfungsi untuk mendeteksi mobil *remote control* yang keluar jalur lintasan ketika sedang berada di lintasan mobil *remote control*.

3.2 Implementasi *Sink Node*

Implementasi *sink node* terdiri dari ESP8266-01, dan kabel *jumper* yang terhubung dengan Arduino Mega 2560 yang menjadi *sink node* (Muiz, Sudiharto, & Putarada, 2019). ESP8266-01 berfungsi sebagai modul *Wi-Fi* agar *sink node* dapat melakukan pengiriman data yang dihasilkan oleh *sensor node* ke *server* melalui protokol TCP/IP sehingga *sink node* dapat terhubung dengan *Wi-Fi* dan dapat terhubung ke internet untuk mengirim dan menyimpan data sensor tersebut ke *web server Thingspeak*. Implementasi *sink node* pada *Single-Hop* WSN lintasan mobil *remote control* dapat dilihat pada Gambar 10.

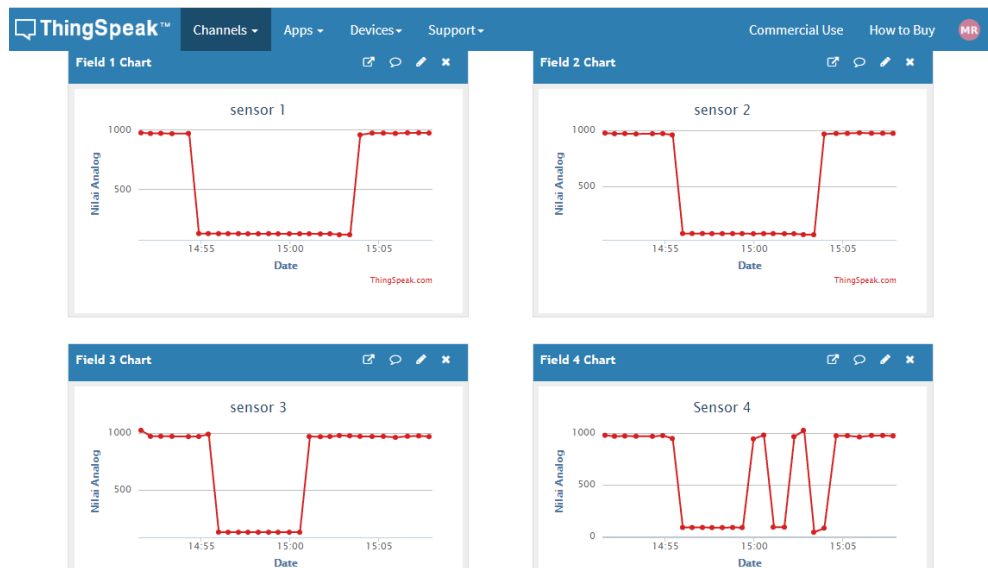


Gambar 10. Implementasi *Sink Node*

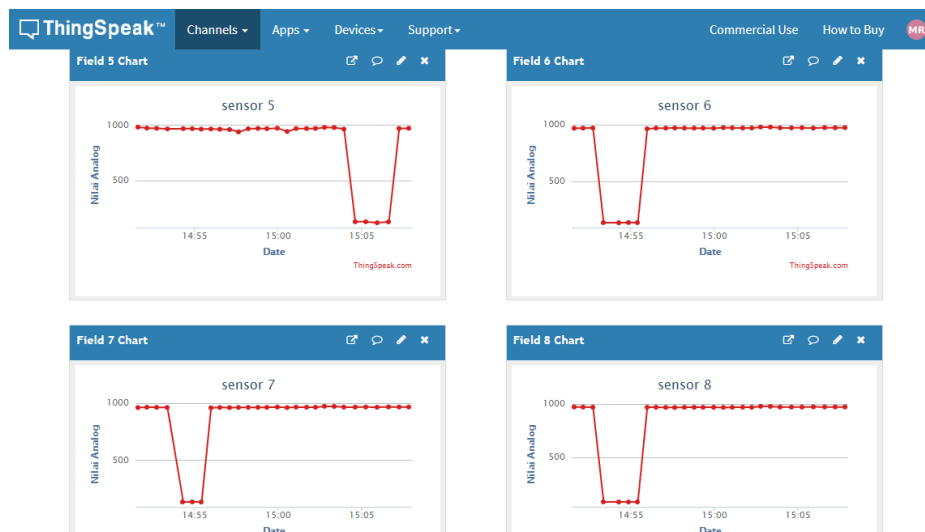
Sink node bertugas menerima data sensor node yang dihasilkan oleh mobil *remote control* ketika berada di lintasan mobil *remote control*. Kemudian data tersebut diteruskan oleh ESP8266-01 yang sudah terhubung langsung dengan *access point* kemudian data diteruskan dengan menggunakan protokol TCP/IP agar data sensor *node* tersebut dikirim menuju ke *web server*. Pada pengujian yang dilakukan untuk mengetahui data sensor yang dikirim oleh *sensor node* dapat ditampilkan di *Thingspeak Server* serta membandingkan data yang dihasilkan, pada sistem ini menerapkan konsep *Internet of Things* (IoT).

3.3 Pengujian Pengiriman Data ke *Server Thingspeak*

Pengujian ini dilakukan untuk mengetahui data *sensor node obstacle avoidance* yang dikirim dapat disimpan serta ditampilkan di *Thingspeak Server* dan mengetahui tiap sensor membaca/tidak adanya objek mobil *remote control* yang sedang dipantau ketika berada di lintasan.



Gambar 11. Hasil Penerimaan Data Pada *Server Thingspeak* Bagian 1



Gambar 12. Hasil Penerimaan Data Pada *Server Thingspeak* Bagian 2

Berdasarkan Gambar 11 dan Gambar 12 merupakan tampilan pada *server Thingspeak* yang

menampilkan 8 buah *Field Chart*, pada tiap *Field Chart* merepresentasikan sensor yang ada pada sistem lintasan mobil *remote control*. Terdapat 28 data yang telah berhasil dikirim kemudian di tampilkan di *server Thingspeak* berupa nilai analog yang dihasilkan oleh *sensor node* yang telah dikumpulkan di *sink node*. Data yang dihasilkan merupakan nilai analog dengan kondisi apabila sensor membaca ada objek didepannya maka nilai analog yang dihasilkan adalah kurang dari 500 dan jika sensor tidak membaca ada objek didepannya maka nilai analog yang dihasilkan adalah lebih dari 500.

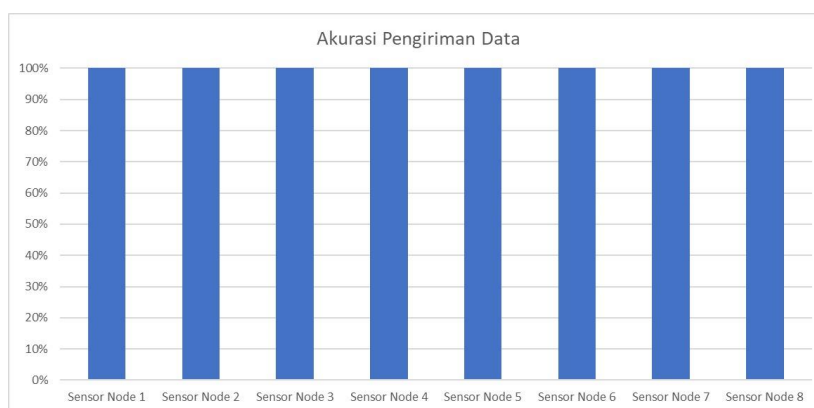
3.4 Pengujian Akurasi Pengiriman Data

Berdasarkan penelitian yang telah dilakukan oleh **(Nashrullah, Primananda, & Widasari, 2018)**. Mengukur tingkat akurasi hasil pengiriman dan penerimaan data pada setiap pengujian yang dilakukan menggunakan Persamaan (1) dan (2) sebagai berikut :

$$Akurasi = \frac{Data\ Yang\ Masuk\ dan\ Benar}{Jumlah\ Total\ Data} \times 100 \quad (1)$$

$$= \frac{400}{400} \times 100\% = 100\% \quad (2)$$

Berdasarkan Gambar 13, dapat dilihat pengujian tingkat akurasi yang telah dilakukan sebanyak 400 kali, yaitu menggunakan 8 buah *sensor node* yang masing-masing pada sensor node diuji sebanyak 50 kali. Jika jumlah data yang dimasukkan dan benar ke dalam web server adalah 400 kemudian dibagi dengan total data yaitu sebanyak 400 kemudian dikalikan dengan 100%, maka hasil tingkat akurasi pengiriman datanya adalah 100%. Hasil perhitungan akurasi pengiriman data tersebut merupakan hasil ketika *sensor node* yang ada pada sistem sedang melakukan pendeteksian pada mobil *remote control* yang sedang melakukan uji di lintasan mobil *remote control* yang kemudian data deteksi tersebut dikirim ke *web server Thingspeak*.



Gambar 13. Grafik Pengujian Akurasi Pengiriman Data

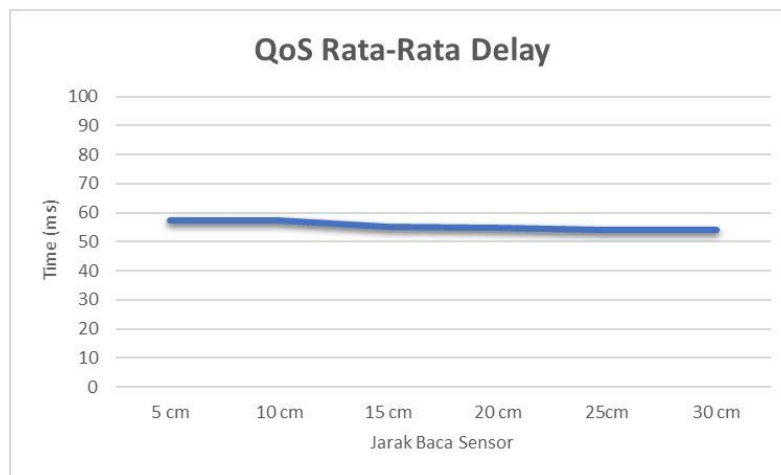
3.5 Pengujian QoS *Single-Hop WSN*

Quality of Service (QoS) merupakan suatu metode untuk dapat mendefinisikan sebuah sifat dan karakteristik pada sebuah *service* dengan cara melakukan pengukuran kualitas jaringan tersebut. QoS biasanya dispesifikasikan untuk mengukur suatu kumpulan atribut kinerja yang diasosiasikan dengan sebuah *service* **(Fahmi, 2018)**. Pengujian QoS dilakukan untuk mengetahui kualitas protokol TCP/IP dalam memberikan layanan komunikasi data dengan mencakup parameter *delay*, *jitter*, *throughput*, dan *packet loss* yang dihitung secara manual **(Sasono, Kusumastuti, Suprianto, Widodo, & Azizcha, 2017)**. Pengujian dilakukan dengan menggunakan variasi jarak baca sensor yang berbeda yaitu, 5cm, 10 cm, 15 cm,

20 cm, 25cm, dan 30 cm, dengan masing-masing diuji sebanyak 15 kali ini dilakukan untuk menguji fungsionalitas *sensor node* ketika proses pengiriman data sedang berlangsung.

Berdasarkan Gambar 14, dapat dilihat bahwa *delay* yang terjadi sangat stabil. Pengujian rata-rata *delay* dilakukan dengan menggunakan protokol TCP/IP ketika sistem sedang melakukan pengiriman data *node* sensor ke *web server*. Hasil pengujian membuktikan bahwa jarak baca sensor tidak mempengaruhi *delay* pada proses pengiriman data yang terjadi dan membuktikan bahwa protokol TCP/IP sangat baik dalam mempertahankan *delay* ketika melakukan transmisi data yang telah diberikan oleh *sensor node* yaitu dengan total rata-rata *delay* sebesar 55ms. Berdasarkan penelitian yang telah dilakukan oleh (Fahmi, 2018), untuk mencari *delay* diperoleh dengan Persamaan (3) sebagai berikut :

$$Delay = \frac{Paket\ Length}{Bandwith} \quad (3)$$

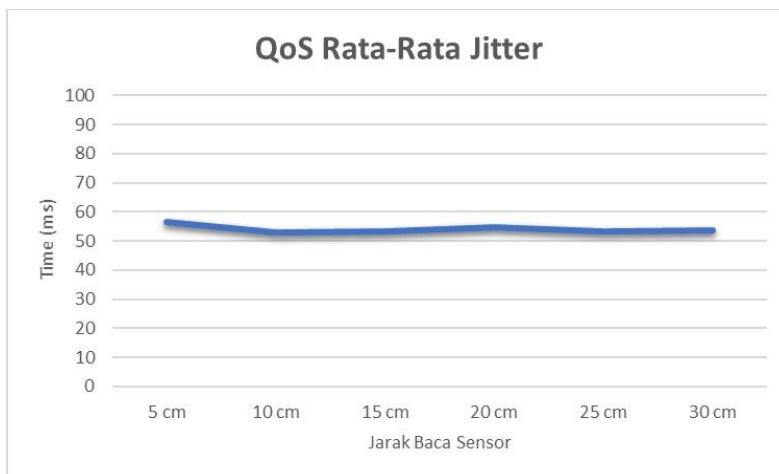


Gambar 14. Pengujian QoS Rata-Rata Delay

Berdasarkan Gambar 15, dapat dilihat bahwa pengujian transmisi data menggunakan protokol TCP/IP hasil *jitter* sangat baik. Pada pengujian ini dilakukan selama 10 detik ketika melakukan transmisi data dengan jarak baca sensor yang berbeda yaitu 5cm, 10cm, 15cm, 20cm, 25cm, dan 30cm. Hasil pengujian membuktikan bahwa jarak baca sensor tidak mempengaruhi *jitter* pada proses pengiriman data yang terjadi dan membuktikan bahwa protokol TCP/IP sangat baik dalam mempertahankan *jitter* ketika melakukan transmisi data yang telah diberikan oleh sensor node yaitu dengan total rata-rata *jitter* sebesar 54 ms. Berdasarkan penelitian yang telah dilakukan oleh (Fahmi, 2018), untuk mencari *jitter* diperoleh dengan Persamaan (4) dan (5) sebagai berikut :

$$Jitter = \frac{Total\ Variasi\ Delay}{Total\ Paket\ diterima} \quad (4)$$

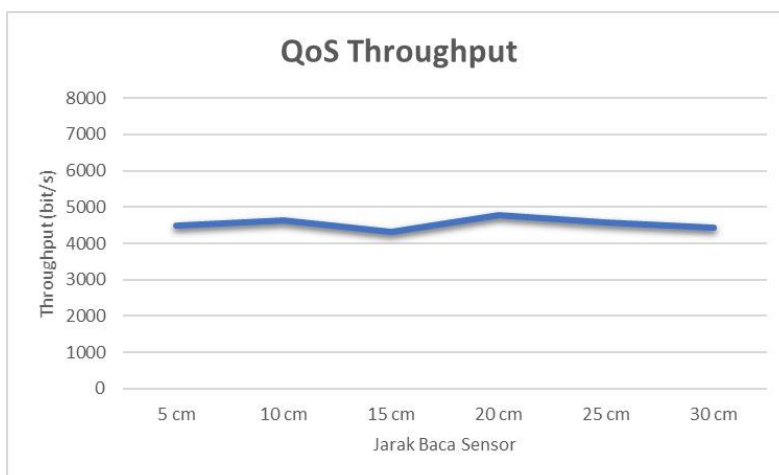
$$Total\ Variasi\ Delay = Delay - rata\ rata\ Delay \quad (5)$$



Gambar 15. Pengujian QoS Rata-Rata Jitter

Berdasarkan Gambar 16, dapat dilihat bahwa pengujian transmisi data menggunakan protokol TCP/IP hasil *throughput* sangat baik dan stabil. Hal ini menunjukkan karena pada pengujian transmisi data, proses transmisi data dilakukan selama 10 detik serta ukuran *packet* yang dikirim berukuran sama melainkan menggunakan variasi jarak baca sensor yang berbeda. Hasil pengujian membuktikan bahwa jarak baca sensor tidak mempengaruhi proses pengiriman data yang terjadi dan membuktikan bahwa *throughput* pada proses pengiriman data hasilnya sangat stabil dengan menggunakan protokol TCP/IP yaitu dengan total rata-rata *throughput* sebesar 4535 bit/s. Berdasarkan penelitian yang telah dilakukan oleh **(Fahmi, 2018)**, untuk mencari *throughput* diperoleh dengan Persamaan (6) sebagai berikut:

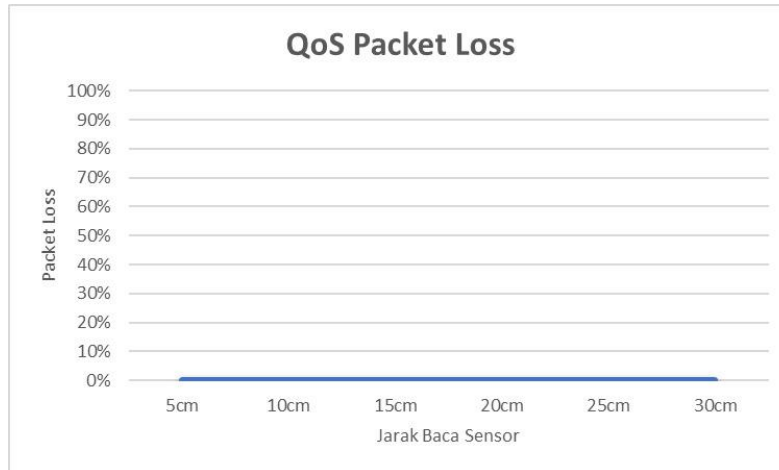
$$Throughput = \frac{Paket\ Data\ diterima}{Waktu\ Pengamatan} \quad (6)$$



Gambar 16. Pengujian QoS Throughput

Berdasarkan Gambar 17, dapat dilihat bahwa pengujian transmisi data menggunakan protokol TCP/IP dapat menangani transmisi paket data dengan sangat baik. Hasilnya tidak ada satu paket pun data yang hilang selama proses transmisi berlangsung berdasarkan variasi jarak baca sensor yaitu 5cm, 10cm, 15cm, 20cm, 25cm, dan 30cm. Berdasarkan penelitian yang telah dilakukan oleh **(Fahmi, 2018)**, untuk mencari *packet loss* diperoleh dengan Persamaan (7) sebagai berikut :

$$\text{Packet Loss} = \frac{\text{Paket Data Dikirim} - \text{Paket Data Diterima}}{\text{Paket Data Dikirim}} \times 100\% \quad (7)$$



Gambar 17. Pengujian QoS Packet Loss

4. KESIMPULAN

Setelah melakukan pengujian dan analisis dari implementasi *Single-Hop Wireless Sensor Network* pada lintasan mobil *remote control* maka diperoleh kesimpulan sebagai berikut :

1. Penelitian ini telah berhasil dirancang dan direalisasikan yaitu implementasi *Single-Hop Wireless Sensor Network* pada sistem lintasan mobil *remote control* dengan menggunakan protokol pengiriman data *Transmission Control Protocol/Internet Protocol* (TCP/IP).
2. Berdasarkan hasil pengujian tingkat akurasi yang telah dilakukan sebanyak 400 kali, sistem ini memiliki tingkat akurasi pengiriman data 100% dan berhasil dikirim ke *server Thingspeak* serta disimpan di *database server Thingspeak*, kemudian jarak baca sensor tidak mempengaruhi QoS selama proses transmisi data ke server.
3. Pada pengujian QoS dengan parameter *delay*, *jitter*, *throughput*, dan *packet loss* hasilnya dapat memberikan layanan jaringan yang baik, berdasarkan hasil parameter *delay*, *jitter*, dan *throughput* yang sangat baik dan stabil, serta tidak adanya paket data yang hilang selama proses transmisi data berlangsung.

DAFTAR RUJUKAN

- Fahmi, H. (2018). Analisis QOS (Quality Of Service) Pengukuran Delay, Jitter, Packet Loss Dan Throughput Untuk Mendapatkan Kualitas Kerja Radio Streaming Yang Baik. *Jurnal Teknologi Informasi Dan Komunikasi*, 98-105.
- Hidayati, N., & Suwadi. (2016). Analisis Kinerja TCP/IP untuk Jaringan Nirkabel Bergerak 3G di Surabaya. *JURNAL TEKNIK ITS*, 941-946.
- Hulu, E., Riyanto T, B., & Widyantoro, S. (2015). Wireless Sensor Network For Volcano Activity. *Scientific Journal of Informatics*, 53-62.
- Muiz, I., Sudiharto, D., & Putarada, A. (2019). Analisis Traffic Pada Implementasi Wireless Sensor Network Polusi Udara. *e-Proceeding of Engineering*, 2048-2056.

- Nashrullah, M., Primananda, R., & Widasari, E. (2018). Implementasi Wireless Sensor Network Pada Keamanan Rumah Menggunakan Sensor Pir. *Jurnal Pengembangan Teknologi Informatika dan Ilmu Komputer*, 7322-7330.
- Nurkamid, M., & Widodo, A. (2021). Penerapan Wireless Sensor Network Untuk Monitoring. *Jurnal IKRAITH-INFORMATIKA*, 72-78.
- Sasono, S., Kusumastuti, S., Suprianto, E., Widodo, S., & Azizcha, D. (2017). QoS Analysis of Wireless Sensor Networks for Temperature and Humidity Monitoring and Control of Soybean Seed Storage Based IOT Using NodeMCU. *Journal of Applied Information and Communication Technologies (JAICT)*, 1-11.
- Subono, Hidayat, A., & Akhmad, A. (2019). Pengaruh Mobilitas End Device Pada Wireless Sensor Network (WSN) Untuk Pemantauan Gas CO dan H₂S Pada Kawah Ijen Kabupaten Banyuwangi. *Jurnal ELTEK*, 12-29.
- Tarmidi, Taqwa, A., & Handayani, A. (2019). Penerapan Wireless Sensor Network Sebagai Monitoring. *Seminar Nasional Inovasi dan Aplikasi Teknologi di Industri 2019*, 224-230.
- Zhao, J., Qiao, C., Sudhaakar, R., & Yoon, S. (2013). Improve Efficiency and Reliability in Single-Hop WSNs with Transmit-Only Nodes. *IEEE Transactions On Parallel And Distributed Systems*, 520-534.