PERBANDINGAN *QUALITY OF SERVICE* PROTOKOL KOMUNIKASI DATA PADA SISTEM DETEKSI ASAP ROKOK BERBASIS *INTERNET OF THINGS*

Pether V.B. Romony, Lanny Sitanayah, Junaidy B. Sanger
Program Studi Teknik Informatika; Fakultas Teknik
Universitas Katolik De La Salle Manado; Kombos-Kairagi 1 Manado, No.Telp: (0431) 871957
e-mail: 15013052@unikadelasalle.ac.id, lsitanayah@unikadelasalle.ac.id, jsanger@unikadelasalle.ac.id

Abstrak*— Asap rokok adalah salah satu asap beracun yang berbahaya bagi kesehatan manusia dari sisi biologis maupun sisi kimiawi. Pada penelitian ini, penulis mengimplementasikan sebuah sistem deteksi asap rokok berbasis *The Internet of Things* menggunakan sensor MQ135, Arduino *board* dan NodeMCU. Kemudian, penulis melakukan perbandingan *Quality of Service* dari dua protokol komunikasi data, yaitu *Transmission Control Protocol* dan *User Datagram Protocol* pada sistem tersebut. Parameter *Quality of Service* yang dibandingkan saat proses pengiriman data adalah *delay* dan *data loss*. Untuk setiap protokol, simulasi dilakukan selama 1 jam dengan pengiriman data setiap 5 detik, 10 detik, sampai 1 menit. Hasil yang diperoleh adalah *data loss* dengan *Transmission Control Protocol* lebih rendah dari pada *data loss* dengan *User Datagram Protocol*, sedangkan *delay* dengan *User Datagram Protocol* lebih rendah dari pada *delay* dengan *Transmission Control Protocol.

***Kata Kunci*— *Delay, Data Loss, Transmission Control Protocol, User Datagram Protocol*.**

I. PENDAHULUAN

Rokok merupakan salah satu olahan dari tembakau yang mengandung zat yang berbahaya bagi kesehatan manusia. Asap yang berasal dari rokok memiliki berbagai bahan beracun yang berbahaya dan bahan kimia yang dapat memicu terjadinya kanker (karsinogen). Berbagai bahan kimia dari rokok bukan hanya dapat mengakibatkan gangguan kesehatan pada orang yang merokok, tetapi dapat membahayakan orang bukan perokok [1].

Dengan pesatnya perkembangan prosesor dan sensor yang berukuran kecil dan murah, saat ini banyak *sensor node* yang digunakan untuk memonitor lingkungan fisik, dengan atau tanpa baterai [2]. *The Internet of Things* (IoT) merupakan sebuah konsep dimana suatu objek memiliki kemampuan untuk bertukar data melalui jaringan Internet tanpa memerlukan interaksi antar manusia atau interaksi manusia dengan komputer [3]. *Quality of Service* (QoS) adalah suatu metode pengukuran yang biasanya digunakan untuk menghitung kualitas sebuah jaringan [4]. Ada dua protokol yang biasa digunakan untuk komunikasi data, yaitu *Transmission Control Protocol* (TCP) dan *User Datagram Protocol* (UDP).

TCP merupakan salah satu protokol yang berorientasi koneksi yang akan menciptakan suatu koneksi virtual antara dua alat dalam mengirim data. Terdapat tiga tahap dalam pengiriman berorientasi koneksi, yaitu pembentukan koneksi, transfer data, dan pemutusan koneksi. Proses dalam pembentukan dan pemutusan koneksi menggunakan mekanisme *three-way handshake*. Berbeda dengan TCP, UDP adalah protokol yang melakukan komunikasi secara sederhana. UDP disebut sebagai protokol tanpa koneksi dan protokol transport yang tidak dapat diandalkan [5].

Pada penelitian [6], sebuah sistem deteksi asap rokok menggunakan Arduino *board* dan Raspberry Pi telah dirancang dan diimplementasikan menggunakan protokol *Message Queuing Telemetry Transport* (MQTT). Sistem seperti ini membutuhkan sebuah protokol untuk mendukung proses pengiriman data yang handal. Oleh karena itu, pada penelitian ini penulis mengimplementasikan protokol komunikasi data pada sebuah sistem deteksi asap rokok berbasis IoT dengan menggunakan sensor MQ135, Arduino *board* dan NodeMCU. Penulis juga melakukan perbandingan QoS untuk parameter *delay* dan *data loss* dari TCP dan UDP untuk mengetahui pengaruhnya saat proses pengiriman data. Parameter *delay* merupakan waktu yang dibutuhkan untuk mengirimkan data dari asal ke tujuan. Sedangkan parameter *data loss* menunjukkan persentase data yang hilang.

II. TINJAUAN PUSTAKA

A. *Quality of Service* (QoS)

QoS adalah suatu metode pengukuran yang digunakan untuk menghitung kualitas sebuah jaringan dan merupakan suatu usaha dalam mendefinisikan atau menggambarkan karakteristik dan model pada sebuah servis [4]. Beberapa parameter yang digunakan untuk mengukur QoS adalah sebagai berikut:

1. *Delay*

   Delay adalah waktu yang diperlukan dalam proses pengiriman data dari asal ke tujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, *congestion* atau waktu proses yang lama.
2. *Data Loss*

   Data loss adalah suatu parameter yang mengukur persentase data hilang yang dapat terjadi karena *collision* dan *congestion* pada jaringan.
3. *Throughput*

   Throughput mengukur kecepatan pertukaran data yang diukur dalam bps *(bit per second)*. *Throughput* adalah jumlah total dari paket yang berhasil dan sukses diterima dalam jangka waktu tertentu dibagi durasi jangka waktu tersebut.
4. *Jitter*

   Jitter disebut juga variasi *delay*, dimana *jitter* berhubungan erat dengan *latency*, yang memperlihatkan banyaknya variasi *delay* pada transmisi data di jaringan. *Jitter* terjadi karena variasi-variasi dalam panjang antrian, dalam waktu pengolahan data, dan juga dalam waktu pembentukan ulang paket-paket di akhir perjalanan.

Pada penelitian ini, hanya akan diukur *delay* dan *data loss*, karena *throughput* biasanya digunakan untuk membandingkan QoS pada jaringan yang berbeda, sedangkan *jitter* biasanya digunakan untuk mengukur variasi *delay* pada data audio dan video.

B. *Transmission Control Protocol* (TCP)

TCP adalah salah satu protokol yang berada pada *transport layer* yang bertugas untuk melakukan komunikasi dari satu komputer ke komputer lain. TCP memiliki kelebihan dalam hal *reliability, byte-stream,* dan *connection-oriented.*

1. *Reliability*
   TCP memiliki beberapa mekanisme dalam pengiriman data, antara lain *checksum, duplicate data detection, retransmission, sequencing,* dan *timers.*
2. *Byte-stream*
   TCP mengirim data dalam urutan-urutan *byte.*
3. *Connection-oriented*
   Sebelum melakukan pengiriman data, TCP membuat koneksi terlebih dahulu ke komputer penerima.

C. *User Datagram Protocol* (UDP)

UDP merupakan protokol yang melakukan komunikasi secara sederhana. UDP sering disebut sebagai protokol tanpa koneksi dan protokol *transport* yang tidak dapat diandalkan [5]. UDP mempunyai sifat-sifat sebagai berikut [7]:

1. *Connectionless*
   Data UDP dikirim tanpa proses negosiasi antar komputer dalam pertukaran informasi.
2. *Unreliable*
   Data UDP dikirim dalam bentuk *datagram* tanpa nomor urut.
3. UDP menyediakan sebuah mekanisme dalam pengiriman pesan ke protokol *application layer* atau proses khusus dalam komputer pada jaringan yang memakai TCP.

D. Penelitian Terkait

Beberapa penelitian sebelumnya telah membandingkan kinerja protokol komunikasi data. Penelitian pada [8] membandingkan TCP Vegas dan TCP New Reno menggunakan antrian *Random Early Detection* dan *Droptail*. Berdasarkan simulasi, diperoleh hasil yang menunjukkan bahwa TCP Vegas lebih unggul dibandingkan TCP New Reno dengan *Random Early Detection*, terutama untuk *packet delivery ratio*, *delay* dan *packet drop*. Untuk *packet delivery ratio*, TCP Vegas menunjukkan hasil yang terbaik pada saat menggunakan *Random Early Detection* dan *Droptail*. Untuk *throughput*, TCP New Reno menunjukkan hasil yang terbaik pada saat menggunakan *Droptail*. Untuk *delay*, TCP Vegas menunjukkan hasil yang terbaik pada saat menggunakan *Random Early Detection*. Sedangkan untuk *packet drop*, TCP menunjukkan hasil yang terbaik pada saat menggunakan *Random Early Detection*.

Penelitian [9] membandingkan UDP dengan *Datagram Congestion Control Protocol* (DCCP) menggunakan data *streaming* multimedia. Berdasarkan hasil pengujian, hasil yang diperoleh DCCP dengan *Congestion Control* ID 2 (CCID2) memiliki kinerja yang lebih baik pada pemanfaatan *bandwidth* dan keadilan untuk VoIP dan data konferensi video, serta *packet loss* untuk data VoIP. Sedangkan DCCP dengan *Congestion Control* ID 3 (CCID3)memiliki kinerja *delay* dan *jitter* yang lebih baik untuk VoIP dan data konferensi video, serta *packet loss* untuk data konferensi video.

Kinerja TCP Vegas dan UDP dibandingkan pada [10] dengan menggunakan data *streaming*. Berdasarkan hasil pengujian yang diperoleh, persentase pemanfaatan *bandwidth* oleh UDP lebih besar dibandingkan TCP Vegas. Sedangkan persentase *packet loss* oleh TCP Vegas lebih tinggi dari UDP. Di samping itu, *latency* dan *jitter* pada TCP Vegas lebih tinggi dibandingkan UDP.

TCP juga digunakan untuk proses pengiriman data yang dapat diandalkan pada sebuah sistem pemantauan irigasi yang terdiri dari *main* dan *field controller* [11]. Pada penelitian ini, dilakukan uji QoS untuk *throughput, packet loss*, dan *delay*.

Berbeda dengan penelitian-penelitian sebelumnya, pada penelitian ini penulis membandingkan QoS dalam hal *packet loss* dan *delay* untuk TCP dan UDP. Data yang dikirimkan adalah data asap rokok dari hasil sensor MQ135.

E. Arduino Uno

Arduino Uno merupakan suatu perangkat elektronik *open source* dengan *chip* mikrokontroler jenis AVR dari perusahaan Atmel sebagai komponen utamanya. Mikrokontroler adalah sebuah *chip* atau IC *(Integrated Circuit)* yang dapat diprogram dengan menggunakan komputer. Mikrokontroler diprogram agar supaya sebuah rangkaian elektronik dapat membaca input, memprosesnya dan menghasilkan output [12].

F. Sensor MQ135

Sensor MQ135 adalah sensor gas yang digunakan untuk mendeteksi kualitas udara. Sensor ini memiliki sensitivitas yang tinggi terhadap amonia, sulfida, asap dan digunakan untuk mendeteksi gas berbahaya lainnya [13]. Sensor MQ135 memiliki empat buah pin, yaitu:

1. Vcc digunakan untuk menyalakan sensor, umumnya tegangan operasi +5V.
2. *Ground* digunakan untuk menyambungkan modul ke *ground* sistem.
3. *Digital out* digunakan untuk mendapatkan output digital dari pin ini, dengan menetapkan nilai ambang dengan potensiometer.
4. *Analog out* yang menghasilkan tegangan analog 0-5V berdasarkan intensitas gas.

G. NodeMCU

Modul WiFi NodeMCU merupakan *firmware* interaktif berbasis e-Lua. NodeMCU dilengkapi dengan *micro* USB *port* yang berfungsi untuk pemrograman atau *power supply.* NodeMCU dilengkapi dengan *push button* yaitu tombol *reset* dan *flash*. NodeMCU selain dapat diprogram memakai bahasa Lua, dapat juga diprogram menggunakan bahasa C menggunakan Arduino IDE [14].

III. METODE PENELITIAN

Metode penelitian dalam penelitian ini menggunakan beberapa tahapan sebagai berikut:

1. Identifikasi masalah dan studi literatur
   Pada tahap ini dilakukan identifikasi masalah yang terjadi untuk menentukan sistem yang akan dibangun. Selain itu

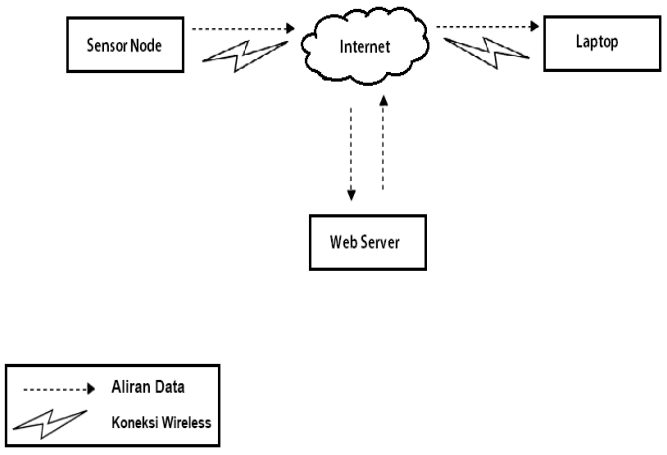
juga, dilakukan studi literatur dari buku dan Internet, seperti *e-book* dan artikel jurnal ilmiah untuk mendukung penelitian yang dilakukan.

2. Analisis
   Melakukan analisis dari penelitian-penelitian sebelumnya dan pengumpulan data yang terkait dengan perancangan sistem.
3. Perancangan
   Pada tahap ini dilakukan perancangan antarmuka, basis data dan alur kerja sistem berdasarkan hasil analisis yang dilakukan pada tahap sebelumnya.
4. Implementasi
   Pada tahap ini dilakukan penerapan terhadap analisis dan perancangan yang telah dilakukan dengan melakukan pemrograman untuk membuat sistem. Sistem yang akan dibangun merupakan sistem berbasis IoT.
5. Pengujian
   Pada tahap dilakukan pengujian terhadap sistem yang telah dibuat dan dilakukan perbandingan QoS dalam hal *delay* dan *data loss* menggunakan TCP dan UDP.
6. Rekomendasi perbaikan
   Pada tahap ini dilakukan pengumpulan rekomendasi perbaikan yang perlu dilakukan berdasarkan hasil pengujian apabila terdapat *error* dalam pembuatan dan pengujian sistem.

IV. HASIL DAN PEMBAHASAN

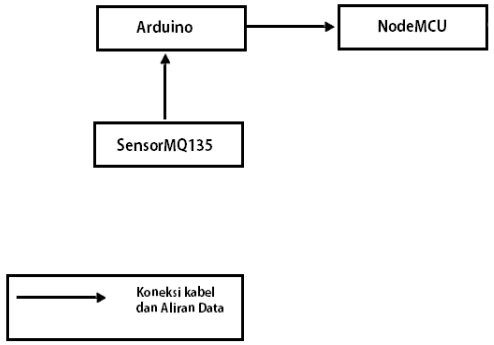
A. Perancangan Sistem

Gambar 1 menunjukkan diagram sistem yang menjelaskan alur dalam sistem deteksi asap rokok berbasis IoT. Pada bagian *sensor node* terdapat beberapa bagian, antara lain Arduino Uno, sensor MQ135 dan NodeMCU. Data yang diperoleh dari *sensor node* akan dikirim ke *web server* dengan menggunakan koneksi Internet. Selanjutnya, data yang ada pada *web server* akan diteruskan menggunakan koneksi Internet ke laptop untuk menampilkan data-data yang berhasil dideteksi oleh *sensor node*.



Gambar 1. Diagram Sistem

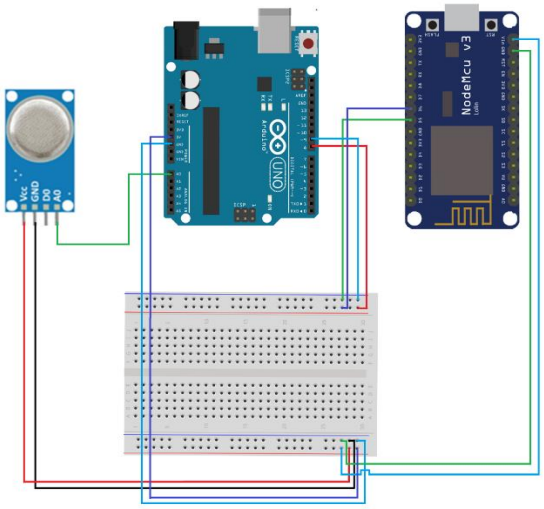
Gambar 2 merupakan diagram *sensor node* yang terdapat pada sistem deteksi asap rokok berbasis IoT, yang terdiri dari Arduino Uno, sensor MQ135 dan NodeMCU. Arduino Uno membaca data dari sensor MQ135 dan meneruskannya ke NodeMCU melalui komunikasi serial. Data tersebut akan dikirimkan ke *web server* menggunakan modul WiFi dari NodeMCU.



Gambar 2. Diagram *Sensor Node*

B. Perancangan Perangkat Keras

Rancangan rangkaian perangkat keras ditunjukkan pada Gambar 3, dimana terdapat sensor MQ135, Arduino Uno *board*, NodeMCU dan sebuah *breadboard* yang dihubungkan dengan menggunakan kabel. Pin A0 pada Arduino terhubung dengan pin A0 pada sensor MQ135, pin GND pada Arduino terhubung dengan pin GND pada sensor MQ135 dan NodeMCU, pin 5V pada Arduino terhubung dengan pin VCC pada sensor MQ135 dan pin Vin pada NodeMCU, kemudian pin 8 pada Arduino terhubung dengan pin D6 pada NodeMCU, dan pin ~9 pada Arduino terhubung dengan pin D5 pada NodeMCU.

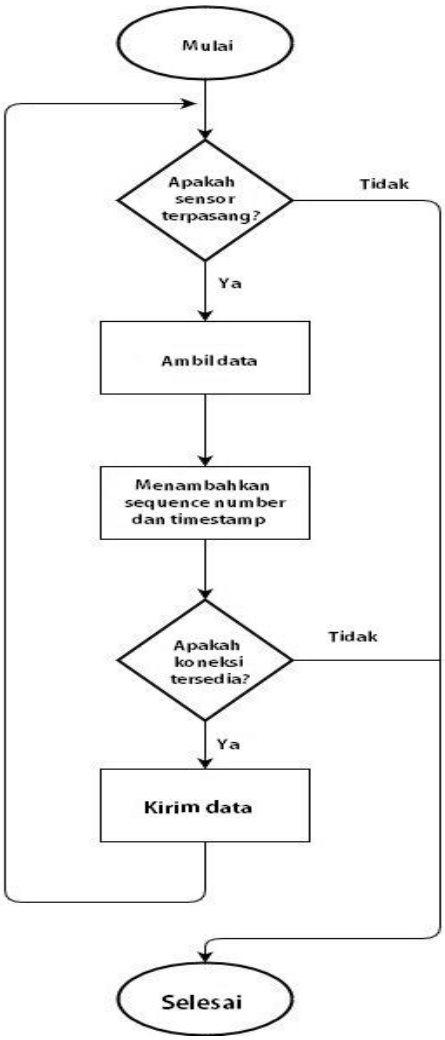


Gambar 3. Perancangan Perangkat Keras

C. Perancangan Perangkat Lunak

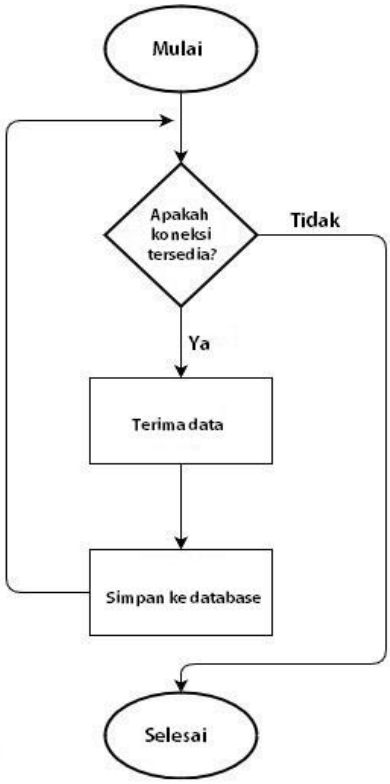
Terdapat dua bagian pada perancangan perangkat lunak, yaitu pada sisi *client* (*sensor node*) dan pada sisi *server*. Gambar 4 adalah *flowchart* yang menggambarkan proses atau alur kerja dari *client* dalam melakukan proses pengiriman data. Pada sisi *client*, langkah pertama yang harus dilakukan yaitu memastikan apakah sensor telah terpasang dengan baik. Jika sensor tidak terpasang dengan baik, maka pengguna tidak dapat menggunakan sistem deteksi asap rokok karena *client* tidak

dapat menerima data dari sensor tersebut. Jika sensor terpasang dengan baik, maka *client* akan menerima data asap rokok yang dideteksi oleh sensor. Proses selanjutnya yaitu *client* akan menambahkan *sequence number* dan *timestamp* ke dalam *payload* berdasarkan data yang terdeteksi oleh sensor. Proses selanjutnya yaitu memastikan bahwa *client* telah terhubung dengan jaringan Internet. Jika tidak terhubung dengan jaringan Internet, *client* tidak dapat mengirim data ke *server*. Jika terhubung dengan jaringan Internet, maka *client* akan mengirim data ke *server*. Proses pada sisi *client* ini akan berjalan terus-menerus.



Gambar 4. *Flowchart client*

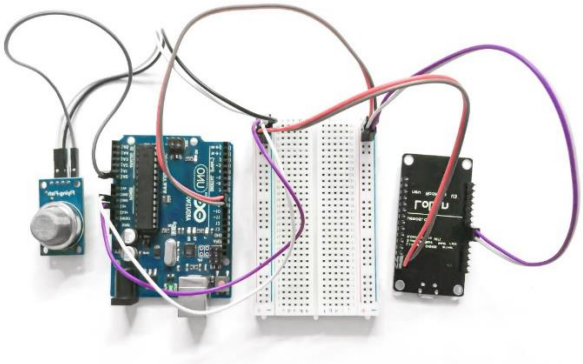
Gambar 5 adalah *flowchart* yang menggambarkan proses atau alur kerja dari *server* dalam melakukan proses penerimaan data. Langkah pertama pada sisi *server* yaitu laptop harus terhubung dengan jaringan Internet. Jika tidak terhubung dengan jaringan Internet, maka *server* tidak dapat menerima data yang dikirim oleh *client.* Jika terhubung dengan jaringan Internet, maka *server* akan menerima data yang dikirim oleh *client*. Proses selanjutnya, *server* akan menyimpan data yang diterima ke dalam *database*. Proses pada sisi *server* ini juga akan berjalan terus-menerus.



Gambar 5. *Flowchart server*

D. Implementasi

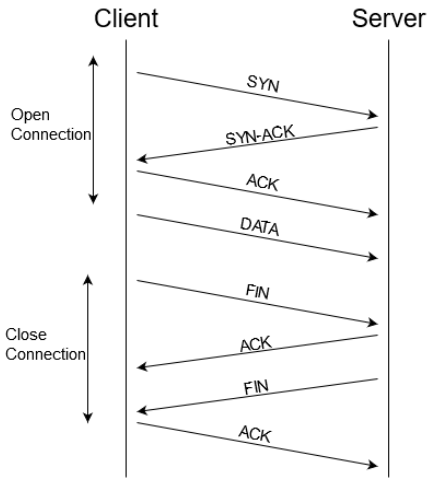
Pada tahap ini, penulis melakukan implementasi perangkat keras dan perangkat lunak yang telah dirancang pada tahap sebelumnya. Implementasi perangkat keras yang terdiri dari sensor MQ135, Arduino Uno *board*, NodeMCU dan sebuah *breadboard* ditunjukkan pada Gambar 6.



Gambar 6. Rangkaian Perangkat Keras

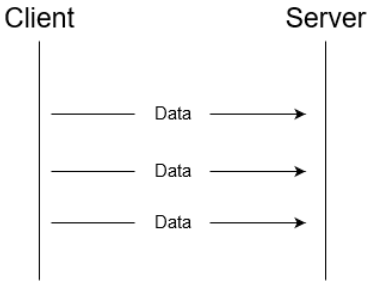
Implementasi dua protokol komunikasi data dilakukan di NodeMCU untuk proses pengiriman data. Gambar 7 menggambarkan tentang prinsip kerja TCP dalam pengiriman data. Sebelum data dikirimkan dari *client* ke *server*, TCP melakukan proses negosiasi (*three-way handshake*) terlebih dahulu untuk membuat koneksi. Tujuannya adalah untuk melakukan sinkronisasi nomor urut dan *acknowledgement* yang dikirimkan antara *client* dan *server*. Proses *three-way handshake* antara *client* dan *server* yaitu pertama-tama *client* akan mengirimkan SYN (*synchronization*) ke *server*. Selanjutnya *server* akan merespon SYN dari *client* dengan

mengirimkan sebuah SYN-ACK (*acknowledgement*) kepada *client*. Setelah *client* menerima SYN-ACK, *client* akan mengirimkan ACK ke *server*. Setelah proses *three-way handshake* tersebut, koneksi baru akan terbentuk dan data dikirimkan dari *client* ke *server*.



Gambar 7. Diagram TCP

Gambar 8 menggambarkan tentang prinsip kerja UDP dalam pengiriman data dari *client* ke *server*. Proses pengiriman data pada UDP yaitu *client* langsung melakukan pengiriman data ke *server* tanpa terlebih dahulu melakukan negosiasi (*three-way handshake*).



Gambar 8. Diagram UDP

Struktur data yang dikirmkan dari *client* ke *server* dengan TCP dan UDP ditunjukkan pada Tabel 1.

Tabel 1. Struktur Data TCP dan UDP

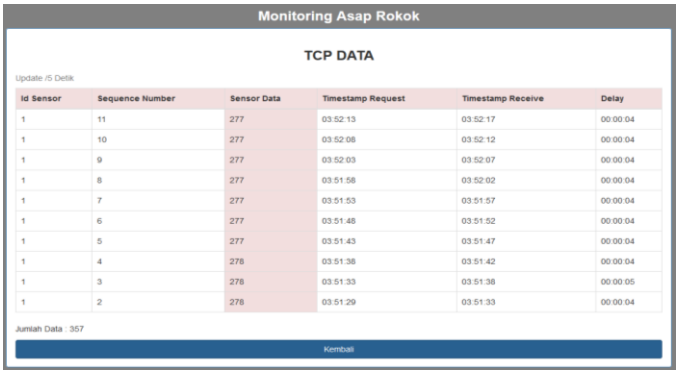
Struktur Data	Tipe Data	Ukuran
ID sensor	Integer	11 Byte
Sequence number	Integer	11 Byte
Sensor data	Integer	11 Byte
Timestamp	Date Time	-

E. Pengujian

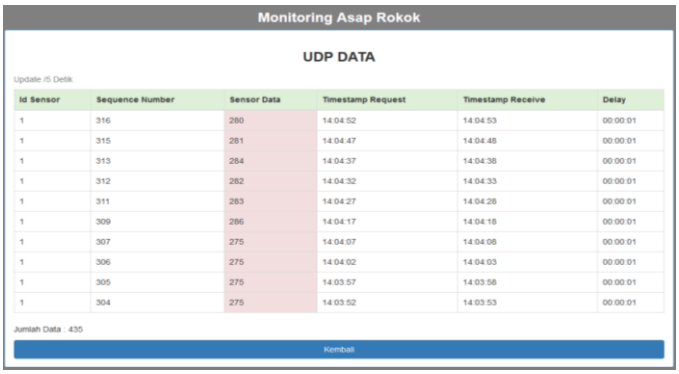
Pada tahap ini dilakukan tiga jenis pengujian terhadap sistem deteksi asap rokok berbasis IoT, yaitu pengujian perangkat keras, pengujian perangkat lunak, dan pengujian protokol komunikasi data, sebagai berikut:

1. Apakah sistem dapat mendeteksi asap rokok?
2. Apakah sistem dapat menampilkan data, seperti ID sensor, *sequence number*, *sensor data*, dan *timestamp*, yang dikirim dari *client* ke *server* menggunakan TCP dan UDP?
3. Bagaimana perbandingan *delay* dan *packet loss* menggunakan TCP dan UDP?

Berdasarkan hasil pengujian perangkat keras dan perangkat lunak, sistem dapat mendeteksi asap rokok dan proses pengiriman data dari *client* menggunakan TCP dan UDP dapat diterima di *server*, disimpan di *database* dan ditampilkan dengan baik. Indikator keberhasilan deteksi asap rokok ditunjukkan dengan data sensor yang berubah ketika terdapat kandungan asap rokok di udara. Tampilan pengujian yang menampilkan data yang dikirim dari *client* ke *server* menggunakan TCP dan UDP ditunjukkan pada Gambar 9 dan Gambar 10.



Gambar 9. Pengujian Menggunakan TCP



Gambar 10. Pengujian Menggunakan UDP

Untuk membandingkan QoS *delay* dan *data loss*, dilakukan simulasi untuk setiap protokol, masing-masing selama 1 jam dengan pengiriman data setiap 5 detik, 10 detik, sampai 1 menit. Tabel 2 menunjukkan perbandingan *data loss* menggunakan TCP dan UDP dari hasil pengujian selama 1 jam, dimana TCP menghasilkan *data loss* rendah dibandingkan UDP. *Data loss* pada UDP bervariasi dikarenakan *traffic* pada jaringan yang bervariasi selama pengujian berlangsung.

Tabel 2. Perbandingan *Data Loss* TCP dan UDP

Sampling Rate	TCP	UDP
5 *second*	0%	34.1%
10 *second*	0%	16.9%
20 *second*	0%	39%
30 *second*	0%	31%
1 *minute*	0%	12%

Tabel 3 menunjukkan perbandingan TCP dan UDP dari pengujian selama 1 jam untuk parameter *delay*. Terlihat pada

tabel bahwa UDP mempunyai *delay* lebih rendah dibandingkan TCP. Hal ini dikarenakan pada proses pengiriman data, UDP tidak melakukan proses *handshake*.

Tabel 3. Perbandingan *Delay* TCP dan UDP

Sampling Rate	TCP	UDP
5 *second*	0.9 ms	0.7 ms
10 *second*	0.06 ms	0.0 ms
20 *second*	0.9 ms	0.5 ms
30 *second*	1.4 ms	0.9 ms
1 *minute*	0.5 ms	0.1 ms

Pada penelitian ini, penulis tidak melakukan perbandingan QoS dengan penelitian-penelitian sebelumnya karena perbedaan perangkat keras yang digunakan dan perbedaan data sensor yang dihasilkan.

V.KESIMPULAN DAN SARAN

Berdasarkan hasil implementasi dan pengujian protokol komunikasi data pada sistem deteksi asap rokok berbasis IoT yang telah dilakukan, maka dapat disimpulkan bahwa sistem dapat berjalan dengan baik, dimana perangkat keras dapat mendeteksi asap rokok dan perangkat lunak dapat mengirimkan dan menerima data menggunakan TCP dan UDP. Nilai parameter *data loss* pada TCP adalah 0%, sedangkan pada UDP bervariasi sesuai dengan *traffic* pada jaringan. Nilai parameter *delay* pada UDP lebih rendah dibandingkan TCP dikarenakan proses *handshake* pada TCP.

Ke depannya, sistem dapat dikembangkan dengan menambahkan *actuator* seperti *alarm* yang diaktifkan ketika asap rokok dideteksi. Juga dapat ditambahkan algoritma *data mining* untuk membedakan asap rokok dan asap-asap lain yang mungkin dideteksi oleh sensor.

DAFTAR PUSTAKA

[1] I. V. D. Batubara, B. Wantouw dan L. Tendean, "Pengaruh Paparan Asap Rokok Kretek Terhadap Kualitas Spermatozoa Mencit Jantan (Mus Musculus)," *Jurnal eBiomedik*, vol. 1, no. 1, pp. 330-337, 2013.

[2] J. Hester, N. Tobias, A. Rahmati, L. Sitanayah, D. Holcomb, K. Fu, W. P. Burleson dan J. Sorber, "Persistent Clocks for Batteryless Sensing Devices," *ACM Transactions on Embedded Computing Systems*, vol. 15, no. 4, pp. 1-28, 2016.

[3] D. Hardika dan N. Nurfiana, "Sistem Monitoring Asap Rokok Menggunakan Smartphone Berbasis Internet of Things (IoT)," *Jurnal Sistem Informasi dan Telematika*, vol. 10, no. 1, pp. 75-82, 2019.

[4] R. Wulandari, "Analisis QoS (Quality of Service) Pada Jaringan Internet (Studi Kasus : Upt Loka Uji Teknik Penambangan Jampang Kulon – Lipi)," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 2, no. 2, pp. 162-172, 2016.

[5] R. T. Y. Anggraeni, Jusak dan A. Sukmaaji, "Analisis Perbandingan Unjuk Kerja Protokol TCP, UDP, dan SCTP Menggunakan Simulasi Lalu Lintas Data Multimedia," dalam *Seminar Teknik Informatika dan Sistem Informasi*, Bandung, 2013.

[6] J. B. Sanger, L. Sitanayah dan V. D. Kumenap, "Detection System for Cigarette Smoke," dalam *The 4th International Conference on Information Technology, Information Systems and Electrical Engineering*, Yogyakarta, 2019.

[7] W. Sulistyo, "Perancangan Reliabilitas Sistem Transmisi Data Pada protokol UDP (User Datagram Protocol)," dalam *Seminar Nasional Informatika*, Yogyakarta, 2009.

[8] G. W. Pamungkas, W. Yahya dan H. Nurwarsito, "Analisis Perbandingan Kinerja TCP Vegas dan TCP New Reno Menggunakan Antrian Random Early Detection dan Droptail," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 10, pp. 3239-3248, 2018.

[9] F. S. Adinata, Jusak dan A. Sukmaaji, "Analisis Perbandingan Unjuk Kerja Protokol UDP dengan DCCP Menggunakan Trafik Data Multimedia," *Journal of Control and Network Systems*, vol. 3, no. 1, pp. 86-93, 2014.

[10] I. R. Andini, Jusak dan S. Anjik, "Analisis Perbandingan Unjuk Kerja Protokol TCP Vegas dan UDP dengan Menggunakan Data Streaming," *Journal of Control and Network Systems*, vol. 3, no. 1, pp. 78-85, 2014.

[11] J. B. Sanger, H. Sukoco dan S. K. Saptomo, "Reliable Data Delivery Mechanism on Irrigation Monitoring System," dalam *International Conference on Advanced Computer Science and Information Systems*, Jakarta, 2014.

[12] E. Ihsanto dan S. Hidayat, "Rancang Bangun Sistem Pengukuran Ph Meter Dengan Menggunakan Mikrokontroller Arduino Uno," *Jurnal Teknologi Elektro*, vol. 5, no. 3, pp. 130-137, 2014.

[13] D. I. Pujiana, A. S. Handayani dan Aryanti, "Perancangan Wireless Sensor Network Dalam Sistem Monitoring Lingkungan," dalam *The 3rd Annual Research Seminar*, Palembang, 2017.

[14] M. F. Wicaksono, "Implementasi Modul WiFi NodeMCU ESP8266 Untuk Smart Home," *Jurnal Teknik Komputer Unikom - Komputika*, vol. 6, no. 1, pp. 1-6, 2017.