

Evaluasi Unjuk Kerja Latency Komunikasi IoT Berbasis UDP untuk Aplikasi pada Intelligent Transportation System (ITS)

Michael Ardita¹⁾, Irmalia Suryani Faradisa²⁾, Bima Romadhon Parada Dian Palevi³⁾

*^{1),2),3)}Teknik Elektro, Institut Teknologi Nasional Malang
Jl. Sigura-gura 2 Malang
Email : michael.ardita@lecturer.itn.ac.id*

Abstrak. Unjuk kerja jaringan nirkabel sangat berpengaruh untuk komunikasi IoT (Internet of Things) yang dipergunakan untuk komunikasi pada suatu ekosistem ITS. Komunikasi pada ITS pada umumnya adalah vehicle to vehicle (V2V) dan vehicle to infrastructure (V2I). Kondisi jaringan nirkabel yang buruk akan dapat membuat aplikasi IoT pada kendaraan terasa berjalan lambat. Untuk dapat mengetahui penyebab unjuk kerja sistem IoT berjalan lambat, maka, pada makalah ini kami mengusulkan untuk evaluasi kinerja jalur komunikasi untuk IoT pada komunikasi V2I dengan berbasiskan protokol komunikasi data UDP (user datagram protocol). Dari hasil pengujian dapat disimpulkan bahwa pada kondisi normal, komunikasi berbasis IoT dengan server yang berada di dalam negeri melalui jaringan selular LTE/4G dapat berjalan dengan cukup cepat yaitu sekitar 70 milidetik. Namun pada kondisi khusus, dimana kondisi jaringan selular sedang padat, komunikasi IoT dengan dapat mendapatkan RTT (round trip time) delay hingga mendekati 60 detik.

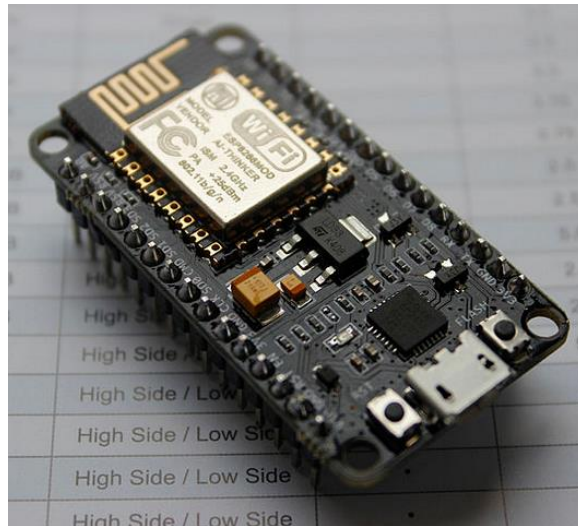
Katakunci: unjuk kerja jaringan nirkabel, IoT, jaringan nirkabel selular, vehicular communication.

1. Pendahuluan

Salah satu pendukung Smart City adalah perangkat Internet of Things (IoT) yang sudah banyak dipergunakan hingga saat ini [1]. Salah satu pilar smart city adalah smart mobility yang salah satu pendukungnya adalah Intelligent Transportation System (ITS). Komunikasi pada ITS secara umum adalah vehicle-to-vehicle (V2V) dan vehicle-to-infrastructure (V2I). Karena V2V saat ini masih belum terbentuk, maka pada penelitian ini difokuskan pada V2I untuk koneksi ke server ITS yang berada pada command center room (CC-room) [2]. Protokol komunikasi data yang dapat digunakan untuk pengiriman data pada ITS dapat menggunakan HTTP (HyperText Transfer Protocol), MQTT (Message Queuing Telemetry Transport), AQMP (Advanced Message Queueing Protocol) dan CoAP (Constrained Application Protocol) pada lapisan aplikasi [3]. Komunikasi pada IoT bisa juga dilakukan dengan langsung mengakses pada protokol pada lapisan transport seperti TCP (Transmission Control Protocol) atau UDP (User Datagram Protocol) [4]. Protokol yang paling mudah diterapkan pada komunikasi IoT adalah HTTP karena layanan server-nya dapat memepergunakan layanan web-hosting yang banyak tersedia di Internet. Sayangnya, pada penelitian sebelumnya, komunikasi berbasis HTTP pada kondisi bergerak dapat menimbulkan latency yang cukup lama apabila jaringan selularnya kurang bagus [5]. Selain itu, dari penelitian sebelumnya juga telah diexplorasi bahwa kepadatan trafik paket data juga dapat menyebabkan latency yang tinggi dalam jaringan nirkabel lokal [6]. Oleh sebab itu, pada penelitian ini kami mencoba untuk membangun komunikasi IoT langsung pada lapisan transport dengan menggunakan protokol UDP. UDP banyak dipergunakan untuk transmisi data yang sifatnya realtime seperti VoIP. Kekurangan UDP adalah kemungkinan terjadinya paket loss karena tidak ada mekanisme retransmisi seperti TCP. Tujuan dari penelitian ini adalah untuk mendapatkan karakteristik komunikasi data berbasis jaringan selular dengan UDP pada kondisi bergerak di area perkotaan.

Pada penelitian sebelumnya telah latency komunikasi timbal balik antara client (On Board Unit – OBU) dan server dengan menggunakan HTTP dapat berlangsung cukup lama apabila kendaraan sedang melintasi area yang layanan Internet lewat jaringan selularnya sedang jelek. Latency pengiriman data bisa mencapai nilai diatas 10 detik apabila sinyal dari jaringan selularnya sedang jelek. Kondisi ini terjadi karena pada komunikasi berbasis TCP ada mekanisme retransmisi paket data apabila dari pengiriman data tidak ada balasan (ACK) dalam kurun waktu 3 detik. Kondisi ini akan berulang beberapa kali hingga sisi pengirim menganggap sudah terjadi kegagalan dalam proses pengiriman data.

Dalam penelitian ini, perangkat yang dipergunakan adalah modul NodeMCU ESP8266 yang sudah dilengkapi dengan modul untuk komunikasi data berbasis WiFi. NodeMCU ini dipilih karena harganya relatif murah (<US\$5) sehingga memungkinkan untuk dikembangkan secara masal [7]. Modul NodeMCU dapat dilihat pada Gambar 1. NodeMCU ini sudah memiliki timer internal dengan resolusi 1 mili detik yang nantinya dapat dipergunakan untuk proses pengukuran unjuk kerja jaringan berbasis selular. Pada NodeMCU ini juga sudah dilengkapi library untuk pengiriman data menggunakan protokol UDP secara langsung.



Gambar 1 – NodeMCU – ESP-8266 [8]

Protokol TCP dan UDP adalah protokol transport yang usianya sudah cukup tua karena lahirnya bersamaan dengan kelahiran Internet. Kelebihan protokol UDP dibandingkan dengan TCP adalah kecepatannya karena tidak terlalu banyak campur tangan kernel dalam menangani paket datanya. Kekurangannya, pengiriman data dengan menggunakan UDP tidak ada garansi bahwa paket data dapat selalu terkirim ke tujuan karena tidak ada proses pengiriman ACK apabila paket data sudah sampai ke tujuan. Kekurangan ini tidak menjadi suatu kendala yang berarti karena proses ACK dapat ditangani juga pada level lapisan aplikasi.

2. Pembahasan

Untuk dapat mendapatkan data unjuk kerja jaringan selular berbasis UDP dapat dilakukan dengan melakukan pengiriman data secara periodik dari OBU menuju ke server. Untuk mempermudah pengujiannya, pengujian dilakukan dengan menggunakan laptop yang dibawa oleh suatu kendaraan. Supaya dapat memperoleh informasi latency pengiriman data, maka paket data yang dikirimkan dari client dilengkapi dengan informasi time-stamp (TTx – time@transmitter) saat pengiriman data. Dari sisi server apabila menerima kiriman data dari client akan memberi balasan ke client dengan melampirkan juga time-stamp pada server (TSvr – time@server) dan juga copy dari TTx. Ketika paket data balasan dari server diterima oleh client, maka akan dapat dihitung waktu latency pengiriman data dari client ke server dan juga balasannya. Karena jam pada client dan server tidak sinkron, maka yang dapat diukur secara pasti adalah selisih waktu antara pengiriman paket data (TTx) dengan waktu penerimaan paket data (TRx – time@receiver). Pada paket data UDP juga dilengkapi dengan id urutan (SeqId) yang nantinya dapat dipergunakan untuk mendeteksi adanya paket loss.

Metode pengukuran performansi jaringan selular dengan menggunakan UDP:

Algoritma pada Client Side UDP-Traffic-Generator :

1. Send data using UDP (SeqId, TTx, ServerAccessKey)
2. Tunggu hingga 10 detik
3. Jika ada pergantian jam dan jumlahLog>100 -> simpan Log dalam bentuk Chart, file text dan SQL
4. Jika tidak ada 'user break' request, kembali ke nomor 1.

Algoritma pada Client-Side.Server-Response-Logger :

1. Dengarkan socket pada port
2. Jika ada data masuk, ambil data TRx, TTx, TSvr, CltId, SeqId, Clt.IPAddr, Clt.Port
3. Hitung RTT dengan cara TRx-TTx.
4. Hitung juga T-C2S=TSvr-TTx dan T-S2C=TRx-TSvr
5. Simpan data SeqId,RTT,TTx,TSvr,SeqId,CltIP untuk analisa lebih lanjut.
6. Tampilkan nilai RTT dalam bentuk grafik batang.
7. Jika jumlah data>JumlahMaximal -> Simpan ke file dan hapus buffer text dan grafik.
8. Jika tidak ada request 'loop-break' dari user kembali ke nomor 1.

Server-Side.Traffic-Responder (menggunakan 2 UDP-Server-Socket):

1. Tunggu sampai ada data UDP masuk, ambil ServerAccessKey,CltID,SeqID dan TTx
2. Jika serverAccessKey=Ok -> Kirim balasan dengan menyertakan TSvr dan Svr.SeqID
3. Jika tidak ada 'user break' request, kembali ke nomor 1.

Adapun paket data yang dikirimkan ke server adalah sebagai berikut :

[Struktur payload UDP] : [SvrAccessKey][CltId][SeqId][TTx]

Adapun paket data balasan dari server adalah sebagai berikut :

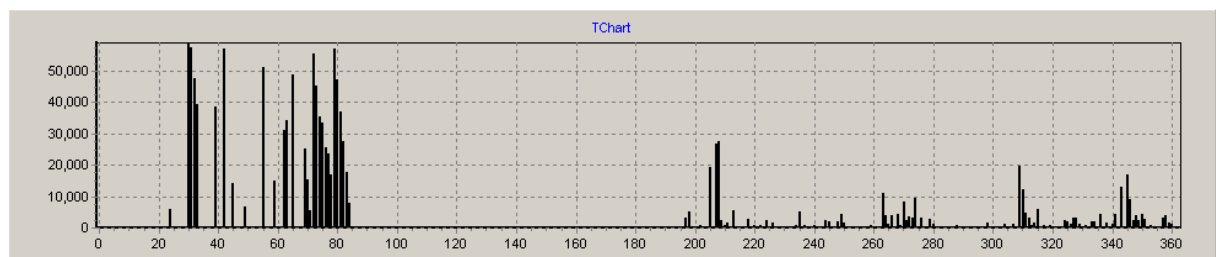
[Struktur payload UDP] : [TSvr][CltId][SeqId][TTx][Clt.IP][Clt.Port]

Proses pengukuran komunikasi IoT dengan protokol UDP ini dilakukan pada 2 lokasi yaitu di area pemukiman pinggiran kota Kediri dan di area perkotaan di kecamatan Lawang dan Singosari, kabupaten Malang. Cuplikan hasil pengukuran yang didapatkan dapat dilihat pada Gambar 2 sampai dengan Gambar 5. Gambar 2 adalah tampilan log data pengukuran dalam bentuk file teks yang berisi data TRx (time-stamp penerimaan), TSvr (time-stamp server), alamat IP client pengirim, dan TTx (time-stamp pengiriman). Gambar 3 dan Gambar 5 adalah grafik hasil pengukuran delay RTT antara client dengan server dimana sumbu X adalah urutan pengiriman dan sumbu Y adalah delay RTT antara client dan server. Gambar 4 dan Gambar 6 adalah data mentah hasil pengukuran dalam format SQL (Structured Query Language) yang nantinya dapat dipergunakan untuk memasukkan data hasil pengukuran ke dalam bentuk database MySQL. Pada log data teks SQL ini juga berisi seperti file teks pada Gambar 2 yang sudah dilengkapi dengan hasil perhitungan delay RTT.

```

Udp_MPMC2022_AII_20220504-105542-Opr2.txt - Notepad
File Edit Format View Help
Listenig to 13722
220504-100225.948|FromSvr:20220504-100225.825|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100220.231|EditFwdKey|25|
220504-100329.341|FromSvr:20220504-100329.202|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100230.232|EditFwdKey|26|
220504-100337.633|FromSvr:20220504-100337.520|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100240.231|EditFwdKey|27|
220504-100337.988|FromSvr:20220504-100337.846|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100250.231|EditFwdKey|28|
220504-100339.676|FromSvr:20220504-100339.526|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100300.233|EditFwdKey|29|
220504-100340.642|FromSvr:20220504-100340.526|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100340.232|EditFwdKey|33|
220504-100438.683|FromSvr:20220504-100438.563|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100400.231|EditFwdKey|35|
220504-100507.359|FromSvr:20220504-100507.245|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100410.231|EditFwdKey|36|
220504-100534.339|FromSvr:20220504-100533.774|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100420.232|EditFwdKey|37|
220504-100616.953|FromSvr:20220504-100616.832|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100510.231|EditFwdKey|42|
220504-100711.516|FromSvr:20220504-100711.332|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100520.232|EditFwdKey|43|
220504-100755.267|FromSvr:20220504-100755.153|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100540.231|EditFwdKey|45|
220504-100821.317|FromSvr:20220504-100821.201|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100550.232|EditFwdKey|46|
220504-100834.491|FromSvr:20220504-100834.379|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100600.231|EditFwdKey|47|
220504-100858.987|FromSvr:20220504-100858.873|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100610.231|EditFwdKey|48|
220504-100935.496|FromSvr:20220504-100935.376|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100710.231|EditFwdKey|54|
220504-100935.572|FromSvr:20220504-100935.458|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100720.232|EditFwdKey|55|
220504-100935.591|FromSvr:20220504-100935.470|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100730.231|EditFwdKey|56|
220504-100935.644|FromSvr:20220504-100935.524|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100740.232|EditFwdKey|57|
220504-100935.717|FromSvr:20220504-100935.604|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100750.232|EditFwdKey|58|
220504-100935.736|FromSvr:20220504-100935.616|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100800.232|EditFwdKey|59|
220504-100943.725|FromSvr:20220504-100943.609|1xx.213.59.130|13722|[Svr.Fwd]>Cmd-Ok|TTx=20220504-100810.231|EditFwdKey|60|
Ln 1, Col 1 100% Windows (CRLF) UTF-8
    
```

Gambar 2 - Log Data Pengukuran UDP di Kota Kediri saat kondisi khusus

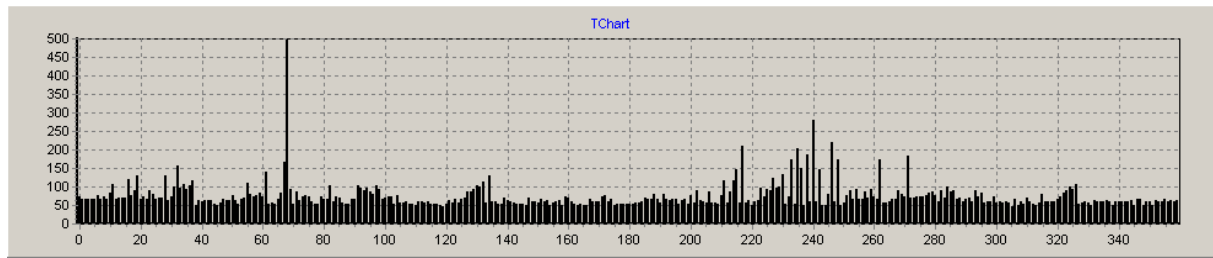


Gambar 3 - Grafik pengukuran delay RTT di kota Kediri pada kondisi khusus

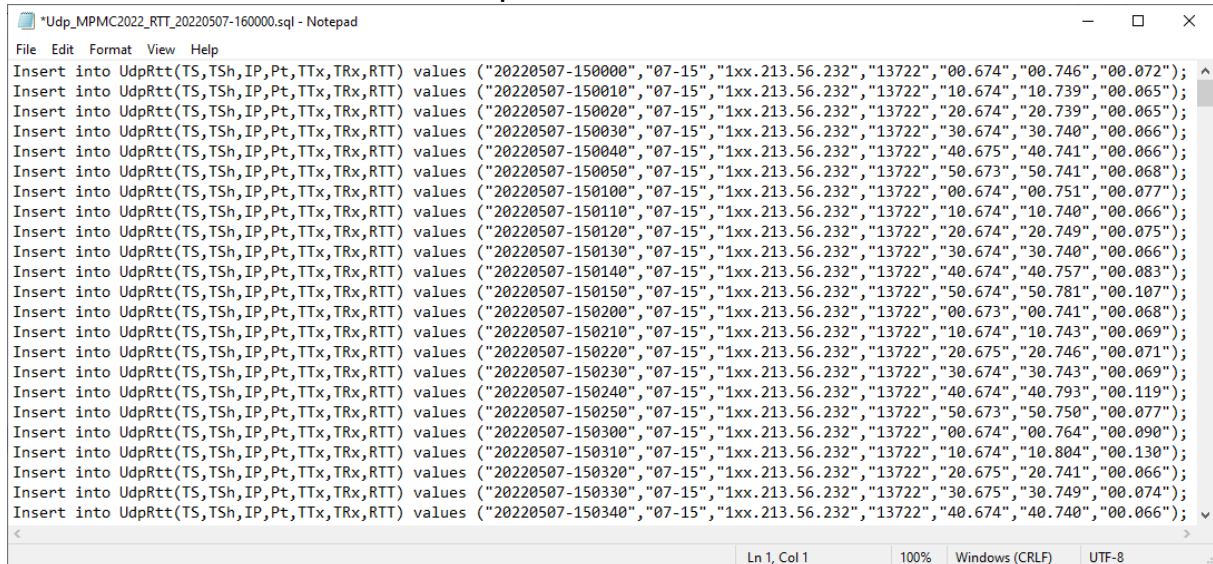
```

*Udp_MPMC2022_RTT_20220504-105542.sql - Notepad
File Edit Format View Help
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100225","04-10","1xx.213.59.130","13722","20.231","25.953","05.722");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100329","04-10","1xx.213.59.130","13722","30.232","29.343","59.111");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100337","04-10","1xx.213.59.130","13722","40.231","37.640","57.409");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100338","04-10","1xx.213.59.130","13722","50.231","37.991","47.760");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100339","04-10","1xx.213.59.130","13722","00.233","39.687","39.454");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100340","04-10","1xx.213.59.130","13722","40.232","40.655","00.423");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100438","04-10","1xx.213.59.130","13722","00.231","38.687","38.456");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100507","04-10","1xx.213.59.130","13722","10.231","07.366","57.135");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100534","04-10","1xx.213.59.130","13722","20.232","34.344","14.112");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100616","04-10","1xx.213.59.130","13722","10.231","16.959","06.728");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100711","04-10","1xx.213.59.130","13722","20.232","11.530","51.298");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100755","04-10","1xx.213.59.130","13722","40.231","55.277","15.046");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100821","04-10","1xx.213.59.130","13722","50.232","21.325","31.093");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100834","04-10","1xx.213.59.130","13722","00.231","34.505","34.274");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100859","04-10","1xx.213.59.130","13722","10.231","58.992","48.761");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100935","04-10","1xx.213.59.130","13722","10.231","35.501","25.270");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100935","04-10","1xx.213.59.130","13722","20.232","35.577","15.345");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100935","04-10","1xx.213.59.130","13722","30.231","35.593","05.362");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100935","04-10","1xx.213.59.130","13722","40.232","35.649","55.417");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100935","04-10","1xx.213.59.130","13722","50.232","35.722","45.490");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100935","04-10","1xx.213.59.130","13722","00.232","35.737","35.505");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100943","04-10","1xx.213.59.130","13722","10.231","43.730","33.499");
Insert into UdpRtt(TS,TSh,IP,Pt,TTx,TRx,RTT) values ("20220504-100945","04-10","1xx.213.59.130","13722","20.232","45.822","25.590");
Ln 1, Col 1 100% Windows (CRLF) UTF-8
    
```

Gambar 4 – Log data pengukuran di kota Kediri dalam format SQL (script database)



Gambar 5 - Grafik pengukuran delay RTT di area urban kecamatan Lawang dan Singosari, kabupaten Malang pada kondisi normal



Gambar 5 - Cuplikan log data pengukuran dalam format SQL di area urban kecamatan Lawang dan Singosari kabupaten Malang pada kondisi normal

Analisa hasil pengukuran

Dari kedua diagram batang (Gambar 3 dan Gambar 5) dapat disimpulkan bahwa pada saat kondisi khusus dan kondisi normal, performansi komunikasi IoT pada kondisi bergerak tampak jauh berbeda. Gambar 3 adalah hasil pengukuran di kota Kediri pada saat masih dalam suasana liburan hari raya Lebaran pada tanggal 4 Mei 2022. Pada kondisi ini, trafik Internet berbasis selular di suatu lokasi bisa meningkat dengan drastis karena banyak pemudik dari kota lain yang pulang kampung. Banyak delay RTT yang nilainya sampai diatas 30 detik. Apabila analisa berdasarkan pada SeqId, ada kemungkinan terjadi banyak packet-loss pada pengukuran tersebut.

Pengukuran berikutnya dilakukan di area kecamatan Lawang dan sekitarnya pada tanggal 7 Mei 2022. Pada pengukuran di kecamatan Lawang, kabupaten Malang terlihat bahwa trafiknya relatif stabil pada RTT delay sekitar 60 msec.

Secara perhitungan, delay RTT antara kecamatan Lawang dengan Server yang berada di Jakarta dapat dihitung sebagai berikut:

$$\begin{aligned}
 \text{RTT delay} &= 2 \times \text{jarak} / (\text{koef} \times c) + \text{delay MAC-CSMA/CA} + \text{delay saat transit pada router dan server} \\
 &= 2 \times 800 \text{ km} / (200,000 \text{ km/detik}) + \text{delay tambahan} \\
 &= (1600 \text{ 000} / 200,000) \text{ msec} + \text{delay tambahan} \\
 &= 8 \text{ msec} + \text{delay tambahan.}
 \end{aligned}$$

Dimana :

koef x c adalah kecepatan rambatan cahaya pada serat optik.

MAC-CSMA/CA adalah Medium Access Control - Carrier Sense Multiple Access with Collision Avoidance

Dari hasil pengukuran dapat ditarik kesimpulan bahwa delay tersebut disebabkan karena adanya proses scheduling untuk akses ke medium pada jaringan selular LTE.

3. Simpulan

Dari hasil pengukuran dan analisa dapat ditarik kesimpulan bahwa :

1. Dengan protokol UDP dapat diamati bahwa komunikasi IoT melalui jaringan selular yang sedang padat, latency pengiriman data ke server hingga mendapatkan balasan dapat mencapai waktu transit yang cukup lama hingga hampir 60 detik apabila menggunakan hotspot dari suatu handphone.
2. Dalam kondisi normal, dengan menggunakan protokol UDP, latency pengiriman data ke server hingga mendapatkan balasan hanya memerlukan waktu sekitar 70 mili-detik.
3. Protokol UDP lebih cepat dibandingkan dengan protokol HTTP yang berbasis UDP karena tidak memerlukan proses hand-shaking terlebih dahulu sehingga cocok untuk aplikasi IoT pada ITS yang memerlukan latency rendah.

Ucapan Terima Kasih

Ucapan terima kasih kami ucapkan kepada ITN Malang yang telah memberikan dana hibah internal untuk penelitian/pengabdian kepada masyarakat.

Daftar Pustaka

- [1]. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, *Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications*, in IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347-2376, Fourthquarter 2015, doi: 10.1109/COMST.2015.2444095.
- [2]. A. Affandi, E. Setijadi, G. Kusrahardjo and M. Ardita, *Data Communication Protocols Supporting Monitoring Service of Traffic Management Center*, 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 2018, pp. 555-559, doi: 10.1109/ISRITI.2018.8864453.
- [3]. N. Naik, *Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP*, 2017 IEEE International Systems Engineering Symposium (ISSE), 2017, pp. 1-7, doi: 10.1109/SysEng.2017.8088251.
- [4]. Fahad Taha AL-Dhief, Naseer Sabri, N. M. Abdul Latiff, Nik Noordini Nik Abd. Malik, Musatafa Abbas Abbood Albader, Mazin Abed Mohammed, Rami Noori AL-Haddad, Yasir Dawood Salman, Mohd Khanapi Abd Ghani, Omar Ibrahim Obaid, *Performance Comparison between TCP and UDP Protocols in Different Simulation Scenarios*, in International Journal of Engineering & Technology, vol. 4 (2018), no. 1, pp. 172-176
- [5]. M. Ardita, Suwadi, A. Affandi and Endroyono, *HTTP communication latency via cellular network for Intelligent Transportation System applications*, 2016 International Conference on Information & Communication Technology and Systems (ICTS), 2016, pp. 34-38, doi: 10.1109/ICTS.2016.7910268.
- [6]. Michael Ardita, Mira Orisa, *WiFi-Based Internet of Things (IoT) Data Communication Performance In Dense Wireless Network Traffic Conditions*, in JEEMECs (Journal of Electrical Engineering, Mechatronic and Computer Science), Vol 4, No 1 (2021), University of Merdeka - Malang.
- [7]. K. J. Singh and D. S. Kapoor, *Create Your Own Internet of Things: A survey of IoT platforms.*, in IEEE Consumer Electronics Magazine, vol. 6, no. 2, pp. 57-68, April 2017, doi: 10.1109/MCE.2016.2640718.
- [8]. Wikipedia-Article, *NodeMCU*, <https://en.wikipedia.org/wiki/NodeMCU> diakses pada tanggal 2022-07-05