



Pengembangan Aplikasi Monitoring Paket Data Sebagai Alat Bantu Perawatan Sistem Persinyalan Kereta Api

Riki Mohammad Taufik Mubarak¹, Ade Rukmana², Irman Nurichsan³

^{1,2,3}Fakultas Teknik Universitas Garut, Garut, Jawa Barat, 44151, Indonesia

Korespondensi: rikimohtaufik19@gmail.com

ARTICLE HISTORY

Received:26-12-2023

Revised:26-06-2024

Accepted:28-06-2024

Abstrak

Kereta api adalah salah satu moda transportasi yang sangat penting dan populer, terutama di Indonesia, dengan jumlah pengguna yang terus meningkat. Dalam operasionalnya, kereta api mengandalkan jaringan komputer yang rumit untuk sistem persinyalan dan kontrol. Namun, seiring dengan perkembangan dan kompleksitas jaringan, kebutuhan akan pemantauan dan pemeliharaan yang efisien semakin meningkat. Terutama, memantau paket data yang mengalir dalam jaringan menjadi kunci dalam menjaga sistem persinyalan berjalan dengan baik. Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi monitoring paket data yang dapat digunakan dalam sistem persinyalan kereta api. Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman Python dan dirancang khusus untuk memantau paket data berbasis protokol UDP yang melintas pada jaringan LAN. Aplikasi ini memungkinkan pengguna untuk memonitor paket data yang dikirim dari sistem kontrol ke objek kontrol / lokasi tujuan, memastikan bahwa sistem persinyalan beroperasi dengan baik. Selain itu, aplikasi juga memfasilitasi pengguna dalam mengidentifikasi masalah dan perubahan status sinyal dengan menggambarkannya dalam bentuk nilai menyala (nilai bit 1) dan padam (nilai bit 0). Hasil pengujian menunjukkan bahwa aplikasi ini mampu berkinerja baik dalam berbagai kondisi jaringan, termasuk jaringan normal, sibuk, dan jarak jauh. Meskipun aplikasi ini memiliki fitur-fitur yang berfungsi dengan baik, penelitian ini juga mengidentifikasi beberapa area perbaikan untuk meningkatkan kinerja dan fungsionalitas aplikasi. Pengembangan aplikasi monitoring paket data ini memiliki potensi untuk meningkatkan efisiensi pemantauan persinyalan kereta api dan dapat menjadi kontribusi berharga dalam pengembangan teknologi pemantauan di sektor transportasi.

Kata kunci: Kereta Api, Sistem Persinyalan, Aplikasi Monitoring, Paket Data, UDP, Jaringan LAN, Python.

Developing Packet Data Monitoring Application As A Tool For Maintaining Railway Signaling Systems

Abstract

Trains are one of the most important and popular modes of transportation, especially in Indonesia, with a continuously growing number of users. In their operations, trains rely on complex computer networks for signaling and control systems. However, as these networks develop and become more complex, the need for efficient monitoring and

maintenance becomes crucial. Particularly, monitoring data packets flowing through the network is key to ensuring the proper functioning of the signaling system. This research aims to develop a data packet monitoring application for railway signaling systems. This application is developed using the Python programming language and is specifically designed to monitor UDP protocol-based data packets traversing a LAN network. The application enables users to monitor data packets sent from the control system to the control object/destination, ensuring the proper functioning of the signaling system. Furthermore, the application facilitates users in identifying issues and changes in signal status by representing them as either on (bit value 1) or off (bit value 0). The test results indicate that the application performs well in various network conditions, including normal, busy, and long-distance networks. Although the application features function effectively, this research also identifies areas for improvement to enhance the performance and functionality of the application. The development of this data packet monitoring application has the potential to enhance the efficiency of railway signaling monitoring and can be a valuable contribution to the development of monitoring technology in the transportation sector.

Key words: *Trains, Signaling System, Monitoring Application, Data Packets, UDP, LAN Network, Python.*

1. Pendahuluan

Kereta api merupakan transportasi yang banyak diminati masyarakat terutama di Indonesia dengan jumlah pengguna yang cukup tinggi [1]. Transportasi ini merupakan salah satu badan usaha milik negara (BUMN) yang dimana aktivitas perkantoran menggunakan jaringan komputer. Kebutuhan user akan kualitas jaringan semakin meningkat baik itu *Local Area Network* (LAN) ataupun *Wide Area Network* (WAN) [2]. Dengan adanya jaringan memudahkan pengguna untuk dapat mengirim ataupun menerima sebuah data dan informasi dalam bentuk paket-paket [3].

Dalam konteks ini, proses pengiriman data menjadi elemen penting yang menghubungkan perangkat-perangkat dalam suatu jaringan [4]. Data tersebut merupakan entitas dasar dari sistem komunikasi yang melibatkan aliran *streamline* terbatas replika lainnya untuk mengirimkan informasi dari satu perangkat ke perangkat lainnya yang di transmisikan melalui lalu lintas jaringan dengan menggunakan router sebagai media pengirimnya [5]. Untuk mengetahui paket data pada suatu jaringan dibutuhkan aplikasi *monitoring* yang dapat bekerja secara *real-time*, dengan tujuan mendapatkan informasi lalu menganalisa paket data yang melintas. Oleh karena itu perlu adanya sistem *monitoring* paket data [6].

Sistem *monitoring* paket data memantau aktivitas lalu lintas jaringan, memastikan stabilitas dan deteksi cepat gangguan [7]. Sistem persinyalan kereta api, sebagai inti sistem, menghubungkan peralatan penting sepanjang rel kereta seperti titik mesin dan lampu sinyal [8]. Untuk menjaga keamanan dan kinerja, perlu implementasi *monitoring* dalam sistem persinyalan. Sistem *monitoring* ini menggunakan jaringan LAN, sehingga sinyal yang di kirimkan melewati jaringan internet yang dapat dimonitoring menggunakan sebuah *software* aplikasi yang dikembangkan menggunakan bahasa pemrograman python.

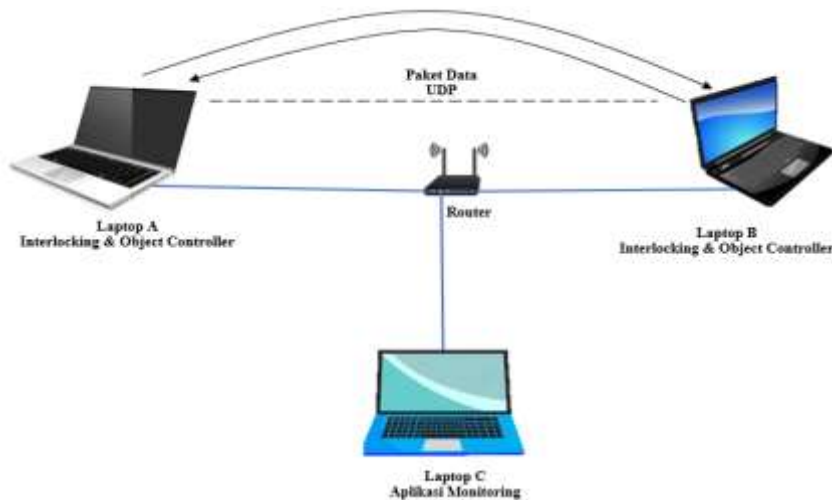
Penelitian sebelumnya, dilakukan oleh Pradana, juga mengembangkan aplikasi *monitoring* paket data [9]. Bedanya, penelitian sebelumnya fokus pada TCP/IP untuk mengawasi jumlah paket data yang masuk dan keluar menggunakan komputer sebagai perangkat pemantauan. Penelitian ini, sebaliknya, difokuskan pada protokol UDP menggunakan bahasa pemrograman Python, dan diimplementasikan pada sistem persinyalan kereta api menggunakan perangkat laptop atau komputer.

Dengan sistem persinyalan kereta api yang krusial, penting memiliki alat untuk memantau alokasi *input/output* pada *object controller*. Aplikasi ini membantu petugas *maintenance* dalam mendeteksi masalah lebih awal dan memantau kinerja secara *real-time*, memastikan sistem berjalan optimal tanpa gangguan yang lebih besar.

2. Metode

Metode penelitian terdiri dari studi literatur, analisis kebutuhan, implementasi, pengujian, dan evaluasi kinerja. Studi literatur melibatkan pencarian dan pemahaman sumber-sumber yang relevan seperti buku, jurnal, dan artikel terkait *monitoring*, bahasa pemrograman Python, serta protokol UDP. Analisis kebutuhan dilakukan untuk merancang perangkat lunak dengan memperhitungkan kebutuhan yang diperlukan. Implementasi dan pengujian dilakukan secara simulasi dengan bertahap untuk memastikan setiap komponen pada aplikasi bekerja dengan baik sebelum diintegrasikan ke sistem. Evaluasi kinerja dilakukan untuk menilai kesesuaian antara rancangan awal dan hasil akhir perangkat lunak.

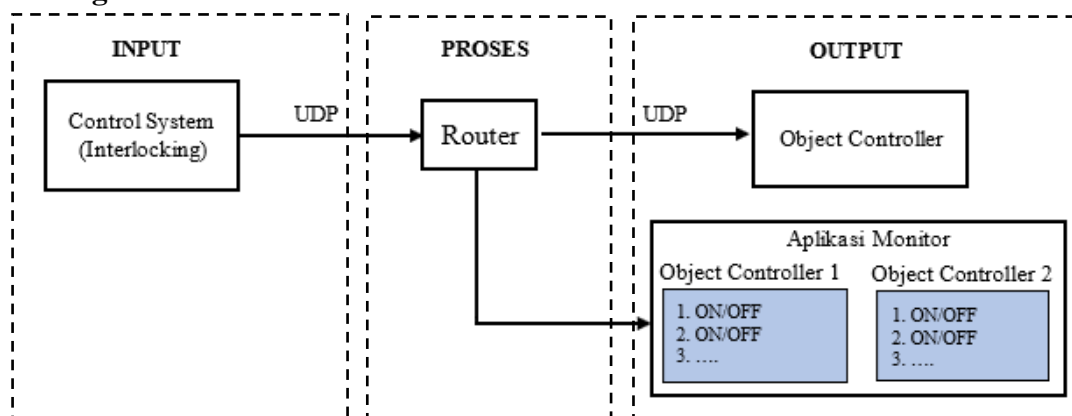
2.1 Skema Simulasi Pengiriman Data



Gambar 1. Skema Simulasi Pengiriman Paket Data

Gambar 1 menunjukkan skema pengiriman paket data menggunakan aplikasi *monitoring* yang diuji melalui simulasi dengan tiga laptop. Laptop A dan B mengirim serta menerima paket data satu sama lain, berfungsi sebagai *interlocking* dan *object controller*. Satu laptop digunakan untuk memantau melalui aplikasi *monitoring*. Jaringan dapat terkoneksi internet atau hanya menggunakan kabel LAN.

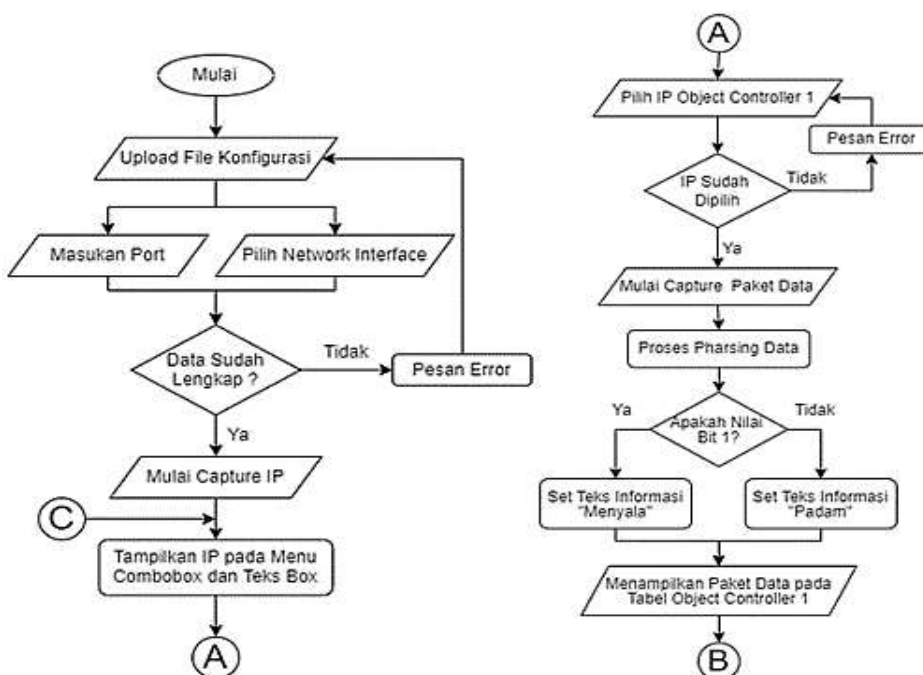
2.2 Diagram Blok

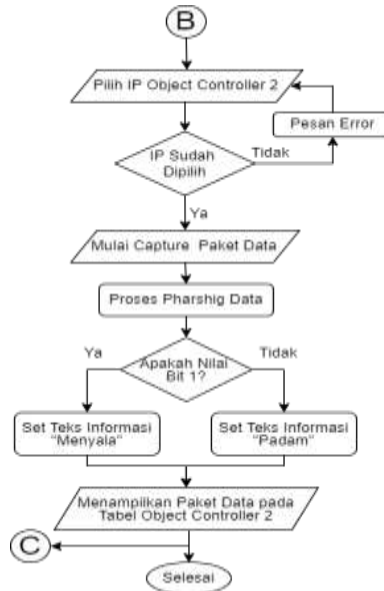


Gambar 2. Diagram Blok

Gambar 2 mengilustrasikan komponen utama dalam sistem persinyalan. Terdapat tiga elemen penting: *input*, *proses*, dan *output*. Bagian *input* menampilkan *Control System (Interlocking)* yang mengatur pengiriman sinyal sebagai paket data. Penelitian ini menggunakan aplikasi simulator yang memiliki fungsi serupa dengan sistem yang diujikan dalam berbagai kondisi jaringan. Data diproses oleh router, bertindak sebagai pengirim, dan disesuaikan dengan kontrol sistem menggunakan protokol UDP. Bagian *output* terdiri dari *object controller* yang menerima paket data untuk mengatur peralatan sistem persinyalan berdasarkan alokasi *input/output* yang telah ditentukan. Aplikasi *monitoring* menampilkan informasi paket data dari *object controller* serta lokasi-lokasi sistem persinyalan secara *real-time*.

2.3 Flowchart Aplikasi





Gambar 3. Flowchart Aplikasi

Dalam aplikasi, prosesnya dimulai dengan inisialisasi program, diikuti oleh langkah-langkah seperti:

- 1) Inisialisasi Program: Sistem siap untuk menerima *input* dan menjalankan proses berikutnya.
- 2) *Upload File* Konfigurasi: Pengguna mengunggah *file* konfigurasi .csv sebagai data referensi yang diperlukan.
- 3) Masukkan *Port & Network Interface*: Pengguna memasukkan nomor port dan memilih antarmuka jaringan.
- 4) Mulai *Capture IP*: Menangkap alamat IP untuk ditampilkan dalam menu *combobox* dan *teks box*.
- 5) Memilih *IP Object Controller 1*: Pengguna memilih alamat IP sebagai lokasi tujuan paket data untuk ditampilkan di tabel *Object Controller 1*.
- 6) Mulai *Capture Paket Data*: Proses capture paket data dimulai.
- 7) Proses *Pharsing Data*: Analisis paket data yang telah ditangkap.
- 8) Perubahan Nilai Bit: Menampilkan nilai bit sebagai teks "menyala" atau "padam" dengan warna yang sesuai.
- 9) Menampilkan Paket Data *Object Controller 1*: Menampilkan hasil *capture* paket data pada tabel *Object Controller 1*.
- 10) Memilih *IP Object Controller 2*: Memilih alamat IP lain untuk melihat lokasi tujuan yang berbeda.
- 11) Mulai *Capture Paket Data*: Memulai lagi proses *capture* paket data untuk IP kedua.
- 12) Proses *Pharsing Data*: Analisis lanjutan terhadap paket data IP kedua.
- 13) Perubahan Nilai Bit: Menampilkan informasi berdasarkan nilai bit.
- 14) Menampilkan Paket Data *Object Controller 2*: Menampilkan hasil *capture* paket data pada tabel *Object Controller 2*.
- 15) Proses Berulang: Proses *capture* paket data dapat diulangi hingga pengguna menghentikannya dengan tombol stop.

Berikut tabel 1 alokasi *input/output* dari sistem asli yang akan dimonitor. Tabel 1 ini merupakan referensi untuk pengiriman sinyal dari *interlocking* ke *object controller*. Pengembangan aplikasi ini memudahkan pemeliharaan persinyalan kereta api dengan memantau secara *real-time*, mengatasi masalah seperti ketidaksinkronan pengiriman sinyal, atau ketidakhadiran sinyal pada *object controller*. Dalam pengembangan aplikasi ini tidak begitu sama dalam menampilkan tabel alokasi *input/output*, bagian *input*-nya tidak di tampilkan secara keseluruhan, namun masih bisa ditambahkan dalam fitur tabelnya agar dapat menampilkan lebih banyak lagi informasi persinyalan, karena

aplikasi ini memakai sistem *upload file*, jadi *file* tersebut dibuat dalam format *.csv*. Dalam tabel alokasi ini ada nama *byte*, *bit*, *name*, *qualifier*, *desc*, mode dan lainnya.

Tabel 1. Alokasi *Input/Output* Persinyalan

MODUL DATA ENTRY FORM										
INTERLOCKING		03CFDGHY	LOC:	F404ZB1	MODULE NO.	3	TYPE:	5		
TEMPLATE		SM2, IM12			POWER SUPPLY:			F404HV		
OUTPUTS						RACK	SLOT	OUTPUT ID	PARAMETER	NOTE
BYTE	BIT	NAME	QUALIFIER	DESC	MODE					
1	7	SGHY782A	R	GE	RR	1	7	8	A_B Arm	
	6	SGHY782A	H	GE	CMD			6	A_B Arm	
	5	SGHY782A	D	GE	CMD			5	A_B Arm	
	4	SGHY782		VR	CMD			4	110 Relay	
	3	SGHY782B	R	GE	RR			7	A_B Arm	
	2							3		
	1							2		
	0	SGHY782C	H	GE	CMD			1	C_Arm	
2	RESERVED									
3	RESERVED									
4	RESERVED									
INPUTS						RACK	SLOT	OUTPUT ID	PARAMETER	NOTE
BYTE	BIT	NAME	QUALIFIER	DESC	MODE					
1	7	F404HV	AC	POK		1	8	8		8B
	6	F404HV	AC	ELD				7		9E
	5							6		
	4							5		
	3							4		
	2	SGHY782		TSSP				3		1D, 2XA
	1	SGHY78		RVP				2		3A
	0	SGHY782N		VP				1		
2	7	F404ZB1/ZB2	NS	ALMK		1	8	12		4B
	6	F404ZB1/ZB3	DC	POK				11		5B
	5	F404ZB1/ZB4	DC	ELD				10		6E
	4	F404ZB2	CHG	ALMK				9		7B
	3									
	2									
	1									
	0									
3	7	SGHY782A	R	(CP)		1	8			
	6	SGHY782A	H	(CP)						

5	SGHY782A	D	(CP)	
4	SGHY782VR		(CP)	
3	SGHY782B	R	(CP)	
2				
1				
0	SGHY782C	H	(CP)	
4	RESERVED			
NOTES:				
1	'SGHY782(TSS) SUPPRESSION FAIL'	7	'F404ZB2 BATTERY CHARGER ALARM'	
2	'SGHY782(TSS) ARMED FAIL'	8	'F404HV AC POWER ALARM'	
3	'SGHY782 TRANSTOP FAIL'	9	'F404HV AC ELD ALARM'	
4	'F404ZB1/ZB2 NETWORK SWITCH ALARM'	10		
5	'F404ZB1/ZB2 DC POWER ALARM'	11		
6	'F404ZB1/ZB2 DC ELD ALARM'	12		

3. Hasil dan Pembahasan

Hasil yang didapatkan berupa hasil dari pengujian aplikasi *monitoring* paket data. Hasil pengujian tersebut akan dibahas dalam tiga konteks yang berbeda, yaitu saat jaringan dalam keadaan normal, saat jaringan mengalami ketidakstabilan atau kesibukan, dan saat pengiriman paket data dilakukan dalam jarak yang cukup jauh. Hasil pengujian dilakukan dengan menggunakan sistem simulasi dengan menggunakan aplikasi simulator yang sudah disediakan untuk percobaan pengiriman data.

3.1 Data Hasil Pengujian Aplikasi *Monitoring* Dalam Keadaan Jaringan Normal

Berikut adalah data hasil yang diperoleh aplikasi *monitoring* dengan melakukan pengiriman paket data menggunakan aplikasi simulator paket data.

NO	SIZE	NAME	QUALITY	STATUS	VALUE
1	7	SGHY782A	H	OK	Paket
2	8	SGHY782A	H	OK	Paket
3	5	SGHY782A	H	OK	Paket
4	4	SGHY782	OK	OK	Paket
5	3	SGHY782B	H	OK	Paket
6	2				Paket
7	2				Paket
8	8	SGHY782C	H	OK	Paket
9	3				Paket
10	7	SGHY782D	H	OK	Paket
11	8	SGHY782D	H	OK	Paket
12	4	SGHY782D	H	OK	Paket
13	4				Paket
14	3	SGHY782E	H	OK	Paket
15	3				Paket
16	4				Paket

Gambar 4. Data Hasil *Monitoring* Kondisi Jaringan Normal

Dari gambar 4 menunjukkan bahwa hasil pengujian dalam keadaan jaringan normal aplikasi *monitoring* berhasil menangkap dan menampilkan paket data dengan baik. Selama kondisi jaringan stabil, aplikasi mampu secara efisien memproses dan menampilkan informasi yang relevan. Berikut tampilan lalu lintas data jika dilihat dari *software* wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
2	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
3	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
4	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
5	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
6	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
7	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
8	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
9	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
10	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
11	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
12	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
13	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
14	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
15	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
16	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
17	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
18	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
19	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
20	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes

Gambar 5. Tampilan Paket Data Kondisi Jaringan Normal

3.2 Data Hasil Pengujian Aplikasi *Monitoring* Dalam Keadaan Jaringan Sibuk

SUBJECT CONTROLLER 1						SUBJECT CONTROLLER 2					
ID	NAME	STATUS	IP	MAC	VALUE	ID	NAME	STATUS	IP	MAC	VALUE
1	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	1	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
2	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	2	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
3	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	3	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
4	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	4	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
5	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	5	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
6	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	6	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
7	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	7	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
8	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	8	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
9	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	9	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
10	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	10	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
11	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	11	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
12	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	12	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
13	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	13	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
14	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	14	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
15	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	15	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
16	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	16	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
17	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	17	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
18	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	18	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
19	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	19	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full
20	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full	20	SGHY760A	OK	192.168.1.1	08:00:27:00:00:00	Full

Gambar 6. Data Hasil *Monitoring* Kondisi Jaringan Sibuk

Gambar 6 menunjukkan bahwa aplikasi *monitoring* berhasil menangkap paket data dengan baik meskipun terjadi peningkatan lalu lintas data yang signifikan. Respons aplikasi tetap relatif cepat dalam memproses dan menyajikan informasi sinyal, bahkan saat ada peningkatan beban jaringan. Ini menunjukkan kemampuan aplikasi dalam menangani situasi dengan lalu lintas data yang meningkat. Jika dilihat dari wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
2	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
3	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
4	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
5	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
6	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
7	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
8	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
9	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
10	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
11	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
12	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
13	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
14	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
15	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
16	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
17	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
18	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes
19	0.000000	192.168.1.10	192.168.1.1	ICMP	60	800000 >> 800000 Echo (ping) 60 bytes
20	0.000000	192.168.1.1	192.168.1.10	ICMP	60	800000 << 800000 Echo (ping) 60 bytes

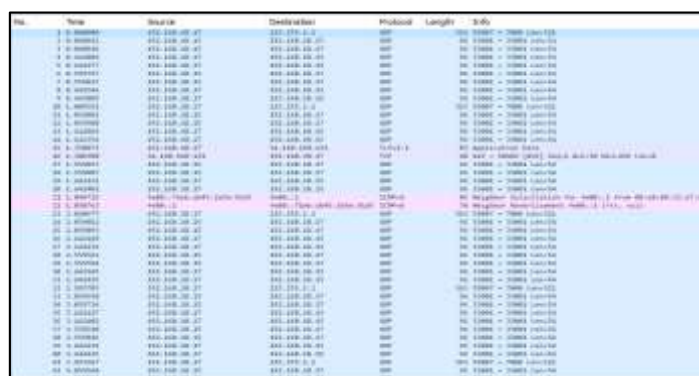
Gambar 7. Tampilan Paket Data Kondisi Jaringan Sibuk

3.3 Data Hasil Pengujian Aplikasi *Monitoring* Dalam Keadaan Jarak Jauh



Gambar 8. Data Hasil Monitoring Kondisi Jaringan Jarak Jauh

Hasil pengujian pada gambar 8, menunjukkan bahwa aplikasi mampu menangkap dan menampilkan paket data dengan baik dalam situasi jarak jauh, sehingga memungkinkan pengguna untuk memantau persinyalan kereta api dengan akurat tanpa memperhatikan jarak pengiriman antara *interlocking* dan *object controller*. Jika dilihat lalu lintas jaringan paket data dari wireshark.



Gambar 9. Tampilan Paket Data Kondisi Jaringan Jarak Jauh

4. Kesimpulan

Dapat disimpulkan dari hasil pengujian menunjukkan bahwa aplikasi ini memiliki performa yang baik dalam mengatasi berbagai kondisi jaringan. Pada jaringan normal, aplikasi mampu menampilkan sesuai dengan yang diharapkan, pada jaringan sibuk dan jarak jauh, aplikasi masih mampu mempertahankan performanya dengan baik. Namun dalam pengujian aplikasi ini masih terbatas, hanya melalui sistem simulasi tidak secara langsung di uji dilapangan. Harapan kedepannya, semoga aplikasi ini dapat berkembang menjadi kontribusi yang berharga dalam pengembangan teknologi pemantauan persinyalan kereta api yang lebih canggih dan andal di masa depan. Selain itu, penelitian ini juga menunjukkan potensi aplikasi sejenis dalam penggunaan di sektor transportasi yang lebih luas.

Ucapan Terima Kasih

Puji dan syukur penulis panjatkan kepada Allah SWT atas karunia dan rahmat-Nya yang melimpah, yang telah memungkinkan penulis menyelesaikan penelitian ini. Serta, penulis mengucapkan terima kasih yang tak terhingga kepada semua pihak yang memberikan dukungan, bantuan, dan motivasi bagi penyelesaian penelitian ini.

Daftar Pustaka

- [1] I. A. Putra, W. W Artha, “Karakteristik Pengguna Moda Transportasi Kereta Api Dan Bus Madiun-Surabaya”, *Jurnal Perkeretaapian Indonesia*, Vol. 4, No 1, pp. 2550-1127, 2020.
- [2] Yusuf. H. F, “Analisis Kebutuhan Dan Kinerja Jaringan Komputer Nirkabel Dengan Parameter Tiphon Di Smk Ma’arif 1 Wates”, *Jurnal Elektronik Pendidikan Teknik Informatika*, Vol. 7 No. 1, 2018.
- [3] B. Paramita, L. A. Abdillah, and E. S. Negara, “Analisis Monitoring Traffic Jaringan Pada PT KAI Divisi Regional Iii Sumsel,” *Fak. Ilmu Komput. Univ. Bina Darma*, no. September, pp. 7–12, 2016.
- [4] F. Afrizala, R. Muzawi, and Y. Efendi, “Analisis Keamanan Lalu Lintas Paket Data Pada Ubuntu Menggunakan Metode Attack Centric,” *Resti (Rekayasa Sist. dan Teknol. Informasi)*, vol. 1, no. 1, pp. 19–25, 2017.
- [5] N. F. Puspitasari, “Implementasi Mikrotik Sebagai Solusi Router Murah Dan Mudah,” no. ISSN : 1978 – 9777, pp. 0–14.
- [6] A. Majid and T. D. Purwanto, “Analisis Dan Monitoring Sniffing Paket Data Jaringan Lokal Bps Sumseldengan Network Analyzer Wireshark,” *Semin. Has. Penelit. Vokasi*, vol. 3, no. 1, pp. 102–109, 2021.
- [7] I. Prasetyo, “Pengenalan Monitoring Jaringan Komputer,” *J. Ilm. Teknol. Inf.*, pp. 0–3, 2013.
- [8] R. Pradana, A. Surya Wibowo, and A. Sugiana, “Perancangan Dan Simulasi Sistem Persinyalan Kereta Api Secara Nirkabel Design and Simulation Railway Signal System Wirelessly,” vol. 8, no. 5, pp. 4397–4408, 2021.
- [9] R. Pradana Wirabadi, “Rancang Bangun Aplikasi Monitoring Paket Data Pada Jaringan,” *175.45.187.195*, no. Maret, p. 31124, 2007