

**ANALISIS *ROBUSTNESS* DAN *RESILIENCE ENTERPRISE*  
*NETWORK* DENGAN *EDGE SENSORS***



Disusun Oleh:

N a m a : Aulia Rahman Hakim  
NIM : 20523061

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA  
FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ISLAM INDONESIA  
2024**

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**ANALISIS *ROBUSTNESS* DAN *RESILIENCE ENTERPRISE*  
*NETWORK* DENGAN *EDGE SENSORS***

**TUGAS AKHIR**



N a m a : Aulia Rahman Hakim  
NIM : 20523061

الجامعة الإسلامية  
Yogyakarta, 6 November 2024

Pembimbing,

( M. Andri Setiawan, S.T., M.Sc., Ph.D. )

## HALAMAN PENGESAHAN DOSEN PENGUJI

**ANALISIS *ROBUSTNESS* DAN *RESILIENCE ENTERPRISE*  
*NETWORK* DENGAN *EDGE SENSORS*****TUGAS AKHIR**

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 6 November 2024

Tim Penguji

M. Andri Setiawan, S.T., M.Sc., Ph.D.

**Anggota 1**

Dr. Syarif Hidayat, S.Kom., M.I.T.

**Anggota 2**

Sheila Nurul Huda, S.Kom., M.Cs.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



( Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D. )

**HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR**

Yang bertanda tangan di bawah ini:

Nama : Aulia Rahman Hakim

NIM : 20523061

Tugas akhir dengan judul:

***ANALISIS ROBUSTNESS DAN RESILIENCE ENTERPRISE  
NETWORK DENGAN EDGE SENSORS***

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 29 Oktober 2024



( Aulia Rahman Hakim )

## HALAMAN PERSEMBAHAN

*Bismillahirrahmanirrahim*

Segala puji syukur penulis panjatkan ke hadirat Allah Swt. atas limpahan rahmat, hidayah, serta karunia-Nya yang telah memberikan kekuatan, kesehatan, dan kesempatan sehingga penulis dapat menyelesaikan tugas akhir ini. Tanpa ridha dan petunjuk-Nya, tugas ini tak mungkin terselesaikan dengan baik. Dengan penuh rasa hormat dan cinta, tugas akhir ini penulis persembahkan kepada:

1. Kedua orang tua tercinta, Ibu Masnani dan Bapak Wagino, yang telah menjadi pilar kekuatan dengan cinta, doa, dan dukungan tiada henti. Terima kasih atas setiap pengorbanan, kasih sayang yang tulus, serta bimbingan yang tiada bertepi. Setiap langkah dalam proses ini berkat semangat yang kalian tanamkan.

2. Bapak Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D., serta Tim *Network Data Center* Badan Sistem Informasi, yang telah memberikan bimbingan, arahan, dan bantuan tak ternilai dalam proses penyusunan tugas akhir ini. Terima kasih atas kesempatan, ilmu, dan waktu yang telah diberikan dengan penuh kesabaran.

3. Sahabat-sahabat seperjuangan, yang selalu setia menemani di setiap langkah, baik dalam tawa maupun tantangan. Terima kasih atas persahabatan yang menguatkan dan semangat yang tak pernah pudar. Bersama kalian, segala hambatan terasa lebih ringan.

4. Diri sendiri, atas usaha, ketekunan, dan kerja keras yang telah dicurahkan dalam proses panjang ini. Semoga tugas ini menjadi langkah awal untuk mencapai segala cita-cita dan impian di masa depan, dengan izin dan ridha-Nya. Aamiin.

Semoga tugas akhir ini dapat memberi manfaat dan menjadi bagian dari perjalanan hidup yang penuh berkah.

**HALAMAN MOTO**

*“Do. Or do not. There is no try”*

(Yoda)

*"In my experience there is no such thing as luck."*

(Obi-Wan Kenobi)

Terlepas apa yang engkau percayai  
Tetap takkan ada yang dibawa mati  
Kembali ke tanah dan tumbuh cemara  
Mana saja harta yang lebih berharga  
(Hindia)

“Menurut saya mimpi...  
tidak ada mimpi yang gagal  
yang ada hanyalah mimpi yang tertunda.  
cuman sekiranya kalau teman-teman merasa gagal dalam mencapai mimpi  
jangan khawatir, mimpi-mimpi lain bisa diciptakan.”  
(Windah Basudara)

## KATA PENGANTAR

*Assalamu 'alaikum Wr. Wb.*

Alhamdulillah rabbil 'alamin, segala puji dan syukur penulis panjatkan kepada Allah Swt. atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad saw., yang telah memberikan syafaat dan membimbing kita keluar dari kegelapan menuju cahaya Islam yang penuh dengan ilmu, iman, dan ihsan. Tugas akhir ini disusun sebagai salah satu syarat untuk menyelesaikan pendidikan di Program Studi Informatika Program Sarjana, Universitas Islam Indonesia. Penulis menyadari bahwa laporan ini tidak akan terselesaikan tanpa bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., selaku Rektor Universitas Islam Indonesia.
2. Bapak Prof. Dr. Ir. Hari Purnomo, M.T., IPU, ASEAN.Eng., selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
3. Bapak Dr. Raden Teduh Dirgahayu, S.T., M.Sc., Ph.D., selaku Ketua Jurusan Informatika, Universitas Islam Indonesia.
4. Bapak Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D., selaku Ketua Program Studi Informatika – Program Sarjana, Universitas Islam Indonesia.
5. Bapak Mukhammad Andri Setiawan, S.T., M.Sc., Ph.D., selaku dosen pembimbing yang dengan sabar membimbing dan memberikan arahan berharga kepada penulis dalam menyelesaikan tugas akhir ini.
6. Seluruh dosen Program Studi S-1 Informatika Universitas Islam Indonesia, atas ilmu, bimbingan, dan pengajaran yang telah diberikan selama masa studi.
7. Network Data Center Badan Sistem Informasi Universitas Islam Indonesia, yang telah membantu penulis dalam pengumpulan data untuk keperluan tugas akhir ini.
8. Kedua orang tua saya, Bapak Wagino dan Ibu Masnani, serta kedua kakak saya, Suhartono dan Dwi Prabawati, yang selalu memberikan dukungan, kasih sayang, dan doa yang tiada henti. Terima kasih atas segala pengorbanan yang telah diberikan.
9. Teman baik saya, Fatimah Azzahra Kusuma Dewi, yang selalu memberikan dukungan dan bantuan selama proses penyusunan tugas akhir ini. Kehadirannya memberikan semangat dan motivasi yang luar biasa.

10. Teman-teman seperjuangan kuliah: Alfito Yoga Pamungkas, Rifaldi Fakhrii, Yuanda Hanif Hisyam, serta teman-teman *Junior Staff* BSI UII angkatan 2020, atas kebersamaan, semangat, dan dukungan yang tak ternilai harganya selama masa studi ini.
  
11. Semua pihak yang telah membantu penulis dalam menyusun tugas akhir ini, yang tidak dapat penulis sebutkan satu per satu. Terima kasih atas segala bantuan dan dukungannya.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang. Semoga laporan ini dapat memberikan manfaat bagi kita semua.

*Wassalamu'alaikum Wr. Wb.*

Yogyakarta, 29 Oktober 2024



( Aulia Rahman Hakim )

## SARI

Penelitian ini bertujuan untuk menganalisis *robustness* dan *resilience* pada jaringan *enterprise* yang rentan terhadap gangguan dengan menggunakan teknologi *edge sensors* berbasis Raspberry Pi 4, Prometheus, dan Grafana. Sistem yang dikembangkan mampu memantau performa jaringan secara *real-time*, termasuk metrik seperti *ping*, *packet loss*, dan kecepatan jaringan. Implementasi Prometheus memungkinkan pengumpulan data metrik yang efisien, sementara Grafana menyediakan dasbor visual yang interaktif untuk menampilkan performa jaringan secara komprehensif.

Hasil penelitian menunjukkan bahwa sistem ini efektif dalam mendeteksi gangguan performa jaringan dan memberikan pemberitahuan dini ketika terjadi penurunan kualitas. Pengaturan ambang batas *ping* dan pengumpulan data berkala membantu menjaga stabilitas jaringan, sementara penggunaan Raspberry Pi sebagai sensor utama mempermudah monitoring dengan biaya rendah.

Dari hasil implementasi dan pengujian, sistem ini mampu memberikan solusi yang efisien untuk memantau kondisi jaringan secara terus-menerus, sehingga meningkatkan *robustness* dan *resilience* pada jaringan. Beberapa rekomendasi untuk pengembangan lebih lanjut adalah integrasi otomatisasi perbaikan jaringan dan penerapan algoritma deteksi anomali berbasis machine learning untuk deteksi gangguan yang lebih akurat.

Kata Kunci: jaringan *enterprise*, *edge sensors*, Raspberry Pi, Prometheus, Grafana, *robustness*, *resilience*, monitoring jaringan.

## GLOSARIUM

<i>Access Point</i>	Perangkat keras yang menghubungkan perangkat nirkabel ke jaringan kabel dan menyediakan koneksi <i>WiFi</i> kepada perangkat pengguna.
Docker	Platform kontainerisasi yang memungkinkan aplikasi dijalankan dalam lingkungan yang terisolasi dan konsisten, mempermudah manajemen aplikasi yang kompleks.
<i>Edge Sensors</i>	Perangkat di tepi jaringan yang mengumpulkan data dan metrik untuk memantau kondisi jaringan secara <i>real-time</i> .
Grafana	Alat visualisasi data <i>open-source</i> yang digunakan untuk membuat dasbor interaktif dan memonitor performa sistem atau jaringan.
<i>Ping</i>	Proses pengiriman paket data dari satu perangkat ke perangkat lain untuk mengukur latensi atau waktu respons jaringan.
<i>Packet Loss</i>	Kondisi di mana paket data yang dikirim melalui jaringan tidak sampai ke tujuan, yang dapat memengaruhi kualitas koneksi jaringan.
Prometheus	Sistem <i>open-source</i> yang digunakan untuk monitoring dan <i>alerting</i> , yang mengumpulkan dan menyimpan metrik performa jaringan dan sistem.
Raspberry Pi 4	Komputer mini yang digunakan sebagai <i>node</i> sensor untuk memantau performa jaringan dengan biaya rendah dan efisiensi tinggi.
<i>Resilience</i>	Kemampuan jaringan untuk tetap berfungsi dengan baik meskipun menghadapi gangguan atau kondisi tidak terduga.
<i>Robustness</i>	Ketahanan sistem jaringan dalam menghadapi perubahan atau gangguan yang tiba-tiba tanpa mengorbankan kinerjanya.
Tailscale	Layanan VPN yang memungkinkan komunikasi <i>peer-to-peer</i> yang aman antar perangkat dengan menggunakan protokol WireGuard.

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING .....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI .....	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR .....	iv
HALAMAN PERSEMBAHAN .....	v
HALAMAN MOTO .....	vi
KATA PENGANTAR .....	vii
SARI .....	ix
GLOSARIUM .....	x
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xiv
DAFTAR GAMBAR .....	xv
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan Penelitian .....	2
1.4 Manfaat Penelitian .....	2
1.5 Batasan Masalah .....	3
1.6 Sistematika Penulisan .....	3
<b>BAB II LANDASAN TEORI</b> .....	<b>4</b>
2.1 Analisis <i>Robustness</i> .....	4
2.2 Analisis <i>Resilience</i> .....	5
2.3 <i>Access Point</i> .....	6
2.4 <i>Edge Sensors</i> .....	6
2.4.1 Raspberry Pi .....	7
2.5 Perangkat Lunak .....	8
2.5.1 Docker .....	8
2.5.2 Prometheus .....	9
2.5.3 Grafana .....	9
2.5.4 Tailscale .....	10
2.6 Penelitian Terkait .....	11
2.6.1 Kesimpulan Penelitian Terkait .....	15
2.6.2 Gap Analisis .....	16
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>17</b>
3.1 Alur Pengerjaan Tugas Akhir .....	17
3.2 Analisis Kebutuhan .....	18
3.2.1 Analisis Permasalahan .....	19
3.2.2 Analisis Fitur Utama .....	19
3.2.3 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak .....	20
3.3 Perancangan .....	22
3.3.1 Instalasi <i>Node</i> Pada Raspberry Pi 4 .....	22
3.3.2 Instalasi Prometheus dan Grafana menggunakan Docker di Windows ...	27
3.3.3 Integrasi Raspberry Pi 4 dengan Prometheus pada Docker di Windows.	30
3.3.4 Penyajian Data Monitoring Access Point pada Grafana melalui Prometheus .....	31
3.3.5 Pengaturan Dasbor dan Penentuan <i>Baseline</i> .....	33
3.3.6 Konfigurasi Tailscale dan UFW Firewall .....	34

3.3.7	Konfigurasi Dasbor Grafana .....	35
3.4	Kesimpulan Metode Penelitian .....	37
BAB IV IMPLEMENTASI DAN HASIL .....		38
4.1	Topologi Pengambilan Data Access Point.....	38
4.2	Penempatan Raspberry Pi 4 .....	39
4.3	Pengumpulan Data .....	39
4.3.1	Pengumpulan Metrik.....	39
4.3.2	Frekuensi Pengambilan Data .....	40
4.3.3	Implementasi Pengumpulan Data .....	40
4.4	Pengujian pada Kondisi Nyata.....	41
4.5	Analisis Variabilitas Ping Menggunakan Standar Deviasi dan Rata-rata Deviasi 42	
4.5.1	Data dan Perhitungan .....	42
4.5.2	Perhitungan untuk 3 September 2024 .....	53
4.5.3	Perhitungan untuk 5 September 2024 .....	54
4.5.4	Perhitungan untuk 12 November 2024 .....	55
4.5.5	Perhitungan untuk 13 November 2024 .....	57
4.5.6	Perhitungan untuk 14 November 2024 .....	58
4.5.7	Perhitungan untuk 15 November 2024 .....	59
4.6	Interpretasi Hasil .....	60
4.7	Perbandingan Waktu Respons <i>Ping</i> pada 3 September 2024 dan 5 September 2024 pada pukul 08:00 – 09:00 .....	61
4.7.1	Data Waktu Respons <i>Ping</i> .....	62
4.7.2	Analisis Performa Jaringan .....	65
4.7.3	Interpretasi Hasil <i>Ping</i> pada 3 September 2024 dan 5 September 2024 pada pukul 08:00 – 09:00.....	65
4.8	Perbandingan Waktu Respons <i>Ping</i> pada 12 hingga 15 November 2024 di Ruangan Kuliah Umum Dr. Sardjito UII .....	66
4.8.1	Data Waktu Respon <i>Ping</i> .....	66
4.8.2	Analisis Performa Jaringan .....	68
4.8.3	Interpretasi Hasil Performa Jaringan pada 12 hingga 15 November 2024 di Ruangan Kuliah Umum Dr. Sardjito UII.....	69
4.9	Perbandingan Performa Jaringan Ruangan Developer BSI UII dan Ruang Kuliah Lantai 1 Gedung Kuliah Umum Dr. Sardjito UII.....	70
4.9.1	Ruang Developer BSI UII.....	70
4.9.2	Ruang Kuliah Lantai 1 Gedung Kuliah Umum Dr. Sardjito.....	71
4.9.3	Analisis Perbandingan.....	72
4.9.4	Kesimpulan Perbandingan Performa Jaringan Ruangan Developer BSI UII dan Ruang Kuliah Lt. 1 Gedung Kuliah Umum Dr. Sardjito UII .....	73
4.10	Pengambilan Data di Ruang Teatrikal Gedung Dr. Sardjito UII.....	73
4.10.1	<i>Ping</i> di Ruang Teatrikal Gedung Dr. Sardjito UII.....	73
4.10.2	<i>Packet loss</i> di Ruang Teatrikal Gedung Dr. Sardjito UII .....	74
4.11	Pengambilan Data di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII .	75
4.11.1	<i>Ping</i> di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII.....	75
4.11.2	<i>Packetloss</i> di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII .....	77
4.12	Mengatur Bot Notifikasi.....	79
4.13	Pengaturan <i>Alert</i> Sistem Monitoring di Grafana Berdasarkan Data Variabilitas <i>Ping</i> .....	80
4.13.1	Analisis Data untuk Penentuan Ambang Batas.....	80
4.13.2	Pengaturan Ambang Batas dan Durasi Pengamatan .....	80

4.13.3	Implementasi <i>Alert</i> di Grafana.....	80
4.13.4	Pemilihan Ambang Batas 30 <i>ms</i> .....	81
4.13.5	Interpretasi dan Penerapan .....	81
4.14	Kesimpulan Implementasi dan Hasil.....	82
BAB V KESIMPULAN DAN SARAN.....		85
5.1	Kesimpulan .....	85
5.2	Saran.....	86
DAFTAR PUSTAKA.....		88
LAMPIRAN .....		90

**DAFTAR TABEL**

Tabel 2.1 Penelitian Terdahulu .....	11
Tabel 3.1 Permasalahan dan dasar fitur utama .....	19
Tabel 4.1 <i>Ping</i> pada tanggal 3, 5 September dan 12, 13, 14, 15 November 2024.....	40
Tabel 4.2 <i>Ping</i> pada 3 September 2024 .....	43
Tabel 4.3 <i>Ping</i> pada 5 September 2024 .....	44
Tabel 4.4 <i>Ping</i> pada 12 November 2024 .....	46
Tabel 4.5 <i>Ping</i> pada 13 November 2024 .....	48
Tabel 4.6 <i>Ping</i> pada 14 November 2024 .....	49
Tabel 4.7 <i>Ping</i> pada 15 November 2024 .....	51
Tabel 4.8 Kinerja jaringan pada tanggal 3 September 2024.....	70
Tabel 4.9 Kinerja jaringan pada tanggal 5 September 2024.....	70
Tabel 4.10 Kinerja jaringan pada tanggal 12 November 2024 .....	71
Tabel 4.11 Kinerja jaringan pada tanggal 13 November 2024 .....	71
Tabel 4.12 Kinerja jaringan pada tanggal 14 November 2024 .....	71
Tabel 4.13 Kinerja jaringan pada tanggal 15 November 2024 .....	72

## DAFTAR GAMBAR

Gambar 2.1 Raspberry Pi 4.....	8
Gambar 3.1 Tahap pengerjaan .....	17
Gambar 3.2 Instalasi <i>Node Exporter</i> .....	23
Gambar 3.3 <i>Prompt</i> mengulang kembali sistem.....	23
Gambar 3.4 Instalasi <i>Ping Exporter</i> .....	24
Gambar 3.5 <i>Prompt</i> mengulang kembali sistem.....	25
Gambar 3.6 Instalasi <i>Speedtest Exporter</i> .....	26
Gambar 3.7 <i>Prompt</i> mengulang kembali sistem.....	27
Gambar 3.8 <i>Prompt</i> mengambil <i>packet</i> Prometheus dan Grafana.....	28
Gambar 3.9 <i>Prompt</i> menjalankan Prometheus .....	28
Gambar 3.10 <i>Prompt</i> menjalankan Grafana .....	28
Gambar 3.11 <i>Config</i> prometheus.yml.....	29
Gambar 3.12 <i>Prompt</i> menjalankan prometheus.yml pada Prometheus.....	29
Gambar 3.13 <i>Prompt</i> menjalankan Prometheus di Docker.....	30
Gambar 3.14 <i>Prompt</i> menjalakan <i>node</i> .....	31
Gambar 3.15 <i>Query ping</i> 8.8.8.8 dan google.com pada Grafana.....	32
Gambar 3.16 <i>Query ping</i> 8.8.8.8 pada Grafana .....	34
Gambar 4.1 Topologi pengambila data <i>access point</i> .....	39
Gambar 4.2 Dasbor Grafana .....	42
Gambar 4.3 Infografis <i>ping</i> pada tanggal 3 September 2024 pukul 08:00 – 09:00.....	62
Gambar 4.4 Infografis <i>ping</i> pada tanggal 5 September 2024 pukul 08:00 – 09:00.....	64
Gambar 4.5 Pemberitahuan <i>ping</i> saat melebihi ambang batas .....	64
Gambar 4.6 Infografis <i>ping</i> pada tanggal 12 November 2024.....	67
Gambar 4.7 Infografis <i>ping</i> pada tanggal 13 November 2024.....	67
Gambar 4.8 Infografis <i>ping</i> pada tanggal 14 November 2024.....	68
Gambar 4.9 Infografis <i>ping</i> pada tanggal 15 November 2024.....	68
Gambar 4.10 Infografis <i>ping</i> di Ruang Teatrikal Gedung Dr. Sardjito UII.....	74
Gambar 4.11 Infografis <i>packet loss</i> di Ruang Teatrikal Gedung Dr. Sardjito UII .....	75
Gambar 4.12 Infografis <i>ping</i> di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII .....	76
Gambar 4.13 Pemberitahuan <i>ping</i> saat melebihi ambang batas .....	77
Gambar 4.14 Infografis <i>packet loss</i> di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII.....	78
Gambar 4.15 Pemberitahuan <i>packet loss</i> saat melebihi ambang batas.....	79
Gambar 4.16 <i>Query ping</i> 8.8.8.8 pada Grafana .....	80



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Teknologi jaringan nirkabel merupakan salah satu perkembangan signifikan dalam dunia komunikasi yang telah merasuki berbagai aspek kehidupan. Dalam teknologi jaringan nirkabel, perangkat komputer dapat saling terhubung melalui sinyal radio atau gelombang udara. Penerapan jaringan nirkabel telah meluas, digunakan baik di lingkungan perkantoran, perumahan, maupun dalam konteks pendidikan.

Di lingkungan universitas, penerapan jaringan nirkabel juga memegang peranan penting. Distribusi internet di universitas dimulai dari *Internet Service Provider (ISP)* yang menyediakan koneksi internet ke universitas. Sinyal ini diterima oleh modem atau perangkat terminasi serat optik yang kemudian diteruskan ke *router* utama. *Router* utama mengatur lalu lintas data dan menghubungkannya dengan *firewall* untuk keamanan jaringan. Dari *firewall*, sinyal diteruskan ke *switch core* yang menghubungkan *router* dengan *switch* distribusi di berbagai gedung. *Switch* distribusi mengalirkan data ke *switch* akses yang berada di dekat pengguna akhir. *Switch* akses ini kemudian menghubungkan perangkat pengguna seperti komputer dan laptop. Terakhir, *access points* menyediakan koneksi *WiFi* kepada pengguna di seluruh area kampus, memastikan semua orang dapat terhubung ke jaringan.

*Access point* di lingkungan kampus sangat penting dalam mendukung berbagai kegiatan, mulai dari proses belajar-mengajar hingga penelitian dan administrasi. Namun, di sisi lain, *access point* juga rentan terhadap berbagai gangguan yang dapat menghambat kinerjanya. Gangguan ini dapat disebabkan oleh berbagai faktor, seperti kerusakan fisik pada perangkat atau kabel, jumlah perangkat yang terhubung terlalu banyak, atau gangguan sinyal akibat kanal yang sama digunakan oleh *access point* lain. Kesalahan konfigurasi juga dapat mengakibatkan performa *access point* menurun, dan serangan keamanan seperti *Denial of Service (DoS)* menambah risiko terhadap keandalan jaringan. Dalam menghadapi tantangan ini, konsep *robustness* dan *resilience* menjadi sangat penting dalam menjaga kualitas jaringan nirkabel.

*Robustness* menggambarkan kemampuan jaringan untuk tetap berfungsi secara optimal meskipun terjadi gangguan atau kondisi yang tidak terduga. Misalnya, jaringan yang *robust* akan mampu mempertahankan kinerjanya ketika jumlah perangkat yang terhubung meningkat atau ketika sinyal terganggu. Sementara itu, *resilience* adalah kemampuan jaringan untuk pulih dengan cepat setelah terjadi gangguan. Ketika gangguan seperti penurunan performa atau

serangan keamanan terjadi, jaringan yang *resilient* mampu kembali berfungsi normal dalam waktu singkat, sehingga meminimalkan dampak gangguan terhadap pengguna.

Untuk menjaga *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus, diperlukan sistem yang mampu memantau performa *access point* secara *real-time* dan memberikan notifikasi dini jika terjadi gangguan. *Edge sensors*, sebagai teknologi monitoring yang canggih, dapat berperan dalam hal ini dengan mendeteksi perubahan performa jaringan dan memberikan informasi kepada administrator untuk segera mengambil tindakan perbaikan. Berdasarkan hal-hal tersebut, penulis melakukan penelitian dengan judul "Analisis *Robustness* dan *Resilience Enterprise Network* dengan *Edge Sensors*." Penelitian ini bertujuan untuk menganalisis pengaruh penggunaan *edge sensors* dalam meningkatkan *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus.

## 1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang di atas, rumusan masalah pada penelitian ini adalah sebagai berikut:

- a. Bagaimana mengetahui *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus?
- b. Bagaimana pengaruh penggunaan *edge sensors* terhadap *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus?

## 1.3 Tujuan Penelitian

Tujuan dilakukan penelitian ini adalah untuk:

- a. Menganalisis penggunaan *edge sensors* terhadap *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus.
- b. Menemukan solusi untuk meningkatkan *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus dengan menggunakan *edge sensors*.

## 1.4 Manfaat Penelitian

Manfaat yang dapat diperoleh dari penelitian yaitu:

- a. Meningkatkan *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus.
- b. Memberikan rekomendasi untuk meningkatkan *robustness* dan *resilience* jaringan nirkabel di lingkungan kampus.
- c. Menjadi referensi bagi penelitian-penelitian selanjutnya yang terkait dengan *robustness* dan *resilience* jaringan nirkabel.

## 1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

- a. Penelitian dilakukan pada lingkungan Universitas Islam Indonesia, Jl. Kaliurang KM 14,5
- b. Data *access point* yang didapatkan hanya di lingkungan Universitas Islam Indonesia, Jl. Kaliurang KM 14,5

## 1.6 Sistematika Penulisan

Sistematika penulisan dokumen skripsi ini adalah sebagai berikut:

### a. BAB I PENDAHULUAN

Bab ini berisi latar belakang penelitian, rumusan masalah, tujuan penelitian, manfaat penelitian, dan batasan masalah yang menjadi dasar dilakukannya penelitian.

### b. BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang mendasari penelitian, seperti konsep mengenai *robustness* dan *resilience* jaringan, teknologi *edge sensors*, Raspberry Pi 4, serta perangkat lunak pendukung seperti Prometheus dan Grafana. Selain itu, bab ini juga mencakup penelitian terdahulu yang relevan.

### c. BAB III METODELOGI PENELITIAN

Bab ini menjelaskan metode yang digunakan dalam penelitian, meliputi analisis kebutuhan, rancangan sistem, serta langkah-langkah implementasi. Metode ini mencakup penggunaan Raspberry Pi 4 sebagai *node* monitoring serta integrasi dengan Prometheus dan Grafana untuk visualisasi data.

### d. BAB IV PEMBAHASAN DAN HASIL

Bab ini menjelaskan proses implementasi sistem yang telah dirancang, termasuk penempatan Raspberry Pi, pengumpulan data metrik, serta hasil dari monitoring yang dilakukan. Analisis performa jaringan juga dibahas berdasarkan hasil yang diperoleh dari dasbor Grafana.

### e. BAB V KESIMPULAN DAN SARAN

Bab terakhir berisi kesimpulan dari hasil penelitian serta saran-saran untuk pengembangan sistem monitoring jaringan ke depannya.

## BAB II LANDASAN TEORI

### 2.1 Analisis *Robustness*

Analisis *Robustness* adalah metode ilmiah yang digunakan untuk menguji sejauh mana layanan dapat diandalkan dan valid dengan menyelidiki variasi kondisi jaringan yang tidak memiliki pengaruh besar pada kualitas layanan, serta dengan mengeksplorasi variasi yang dapat memengaruhi kinerja layanan (Jones, 2021). Tujuannya adalah untuk menentukan apakah kualitas layanan tetap konsisten ketika dikenai berbagai kondisi operasional yang berbeda atau apakah layanan tersebut sangat sensitif terhadap perubahan-perubahan tersebut. Dengan melakukan analisis semacam ini, pengelola jaringan dapat memastikan bahwa layanan yang disediakan oleh *access point* tetap stabil dan andal, serta memenuhi standar yang diharapkan oleh pengguna di bawah berbagai kondisi jaringan.

*Robustness* yang dimaksud di sini adalah kemampuan jaringan untuk tetap berfungsi secara optimal meskipun menghadapi berbagai gangguan atau kondisi yang tidak terduga. Hal ini berarti bahwa jika jaringan mengalami masalah, terdapat metrik-metrik yang dapat digunakan untuk mengidentifikasi sumber masalah dan menilai tingkat keparahan gangguan tersebut. Misalnya, metrik seperti *ping*, *packet loss*, kecepatan unduh dan unggah, serta *jitter* dapat memberikan indikasi awal tentang kesehatan jaringan.

Dengan memantau metrik-metrik ini secara rutin, penanggung jawab pengelola jaringan dapat segera mengetahui adanya anomali atau degradasi performa. Jika metrik menunjukkan adanya masalah, tindakan segera perlu diambil untuk memperbaikinya. Langkah-langkah perbaikan bisa meliputi penyesuaian konfigurasi jaringan, peningkatan kapasitas perangkat keras, atau investigasi lebih mendalam untuk menemukan akar penyebab masalah.

Dengan menindaklanjuti masalah jaringan secara proaktif berdasarkan metrik-metrik yang dikumpulkan, jaringan dapat dijaga agar tetap *robust*. Artinya, jaringan akan lebih tahan terhadap berbagai kondisi yang dapat menyebabkan gangguan, dan dapat kembali beroperasi dengan normal dalam waktu singkat setelah mengalami masalah. Pendekatan ini tidak hanya memastikan kualitas layanan yang konsisten bagi pengguna, tetapi juga mengurangi waktu henti dan meningkatkan efisiensi operasional secara keseluruhan.

## 2.2 Analisis Resilience

Analisis *resilience* adalah proses untuk menilai dan mengevaluasi kemampuan suatu sistem untuk melawan, beradaptasi, dan pulih dari peristiwa atau gangguan yang mengganggu. Ini melibatkan analisis berbagai faktor yang berkontribusi terhadap ketangguhan suatu sistem dan memahami bagaimana faktor-faktor ini berinteraksi untuk menentukan tingkat ketangguhan keseluruhan sistem. Analisis *resilience* merupakan langkah penting dalam manajemen ketangguhan, karena membantu mengidentifikasi kerentanannya, menilai risiko, dan mengembangkan strategi untuk meningkatkan ketangguhan suatu sistem (Mottahedi et al., 2021).

*Resilience* yang dimaksud di sini adalah kemampuan sistem untuk memberikan pemberitahuan dini tentang masalah yang terjadi pada *access point* melalui saluran komunikasi seperti Telegram. Misalnya, jika *access point* pada Gedung Abdul Kahar Muzakir Universitas Islam Indonesia yang dipantau melewati ambang batas yang telah ditetapkan pada grafik, sistem akan memberikan notifikasi melalui Telegram untuk memberitahu penanggung jawab bahwa *access point* tersebut bermasalah. Dengan menerima notifikasi ini, penanggung jawab dapat segera melakukan pengecekan awal pada grafik yang tersedia untuk mengidentifikasi adanya anomali atau gangguan yang terdeteksi. Jika anomali terkonfirmasi, penanggung jawab dapat segera mengambil langkah-langkah yang diperlukan untuk memeriksa dan memperbaiki *access point* yang bermasalah.

Pendekatan ini memastikan bahwa masalah dapat diidentifikasi dan ditangani dengan cepat, sehingga mengurangi waktu henti jaringan dan memastikan kelancaran operasional. Implementasi sistem notifikasi ini juga merupakan bagian penting dari strategi manajemen ketangguhan, karena memungkinkan penanganan masalah secara proaktif dan efisien. Dengan demikian, sistem jaringan dapat tetap *resilient* dan mampu memberikan kinerja yang stabil dan andal meskipun menghadapi berbagai gangguan atau kondisi yang tidak terduga.

Setelah membahas pentingnya *robustness* dan *resilience* dalam menjaga kualitas layanan jaringan, kita dapat mengalihkan fokus ke perangkat kunci yang memainkan peran sentral dalam infrastruktur jaringan: *Access Point (AP)*. *Robustness* dan *resilience* yang kita bahas sebelumnya adalah atribut yang sangat penting untuk dipertimbangkan dalam pemilihan dan pengelolaan *access point*, karena perangkat ini merupakan penghubung utama antara perangkat pengguna dan jaringan kabel.

### 2.3 Access Point

*Access point (AP)* adalah perangkat keras jaringan yang memungkinkan perangkat yang dapat terhubung secara nirkabel untuk terhubung ke jaringan kabel (Xu et al., 2010). Ini adalah perangkat yang menciptakan jaringan area lokal nirkabel (*WLAN*), biasanya di kantor atau bangunan besar, dan terhubung ke *router* kabel, *switch*, atau *hub* melalui kabel *Ethernet*, memancarkan sinyal *WiFi* ke area yang ditentukan. Sebuah AP bisa menjadi perangkat mandiri dengan koneksi kabel ke *router*, atau bisa menjadi komponen integral dari *router* nirkabel.

*Access point* dapat menangani banyak koneksi simultan masing-masing, dan dengan menginstal *access point* di seluruh kantor, pengguna dapat berpindah dari ruangan ke ruangan tanpa mengalami gangguan jaringan (Khairullah, 2024). Ketika perangkat nirkabel bergerak di luar jangkauan satu AP, maka akan dialihkan ke AP berikutnya (Kelly, 2022).

*Access point* ini menjadi objek utama dalam penelitian ini, di mana pemantauan akan dilakukan pada salah satu *access point* di UII. Penelitian ini bertujuan untuk memantau kinerja *access point* tersebut menggunakan berbagai metrik seperti kecepatan unduh, kecepatan unggah, latensi, dan *jitter*.

Data yang dikumpulkan akan dianalisis dan ditampilkan melalui dasbor yang telah dibuat menggunakan Grafana. dasbor ini akan menampilkan grafik yang memperlihatkan performa *access point*, memungkinkan pengelola jaringan untuk dengan mudah mengidentifikasi masalah yang terjadi dan mengambil tindakan yang diperlukan. Dengan adanya dasbor ini, pengelola jaringan dapat memantau kondisi *access point* secara efisien dan memastikan kualitas jaringan yang optimal di lingkungan UII.

Setelah memahami peran krusial *access point* dalam infrastruktur jaringan, kita beralih ke teknologi yang mendukung pemantauan dan analisis kinerja jaringan, yaitu *edge sensors*. *Edge sensors* memungkinkan pemantauan yang lebih detail dan *real-time*, sehingga memperkuat pendekatan *robustness* dan *resilience* yang telah dibahas sebelumnya.

### 2.4 Edge Sensors

*Edge Sensors* adalah perangkat yang digunakan untuk mendeteksi atau mengendalikan aliran data di batas antara dua jaringan. Mereka memenuhi berbagai peran tergantung pada jenis perangkat yang digunakan (Hannats et al., 2021). Edge sensor di sini adalah perangkat keras yang digunakan untuk memonitor *access point* atau bisa disebut *node*. *Node* ini nantinya akan memonitoring *access point* yang menjadi objek utama dalam penelitian ini. Dengan memanfaatkan *edge sensors*, kita dapat mengumpulkan data penting dari *access point* dan

memantau kinerjanya secara *real-time*. Data yang dikumpulkan oleh *edge sensors* akan digunakan untuk analisis lebih lanjut dan ditampilkan pada dasbor yang telah dirancang, sehingga memudahkan pengelola jaringan untuk melakukan pemantauan dan tindakan yang diperlukan.

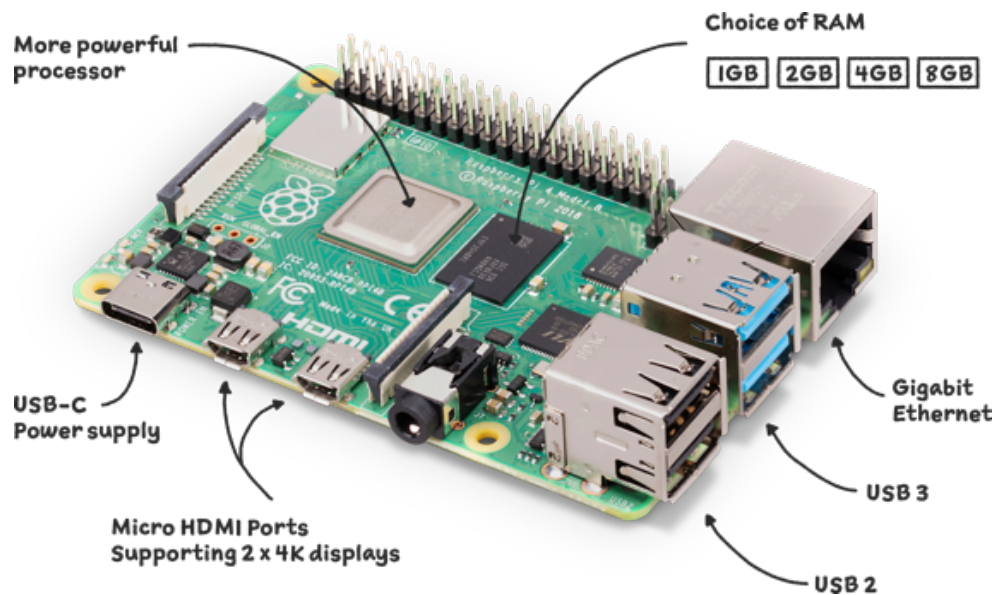
Setelah membahas peran penting *edge sensors* dalam pemantauan dan pengendalian jaringan, penting untuk menyadari bahwa perangkat keras yang digunakan untuk mendukung fungsi ini harus memiliki spesifikasi yang cukup untuk menangani tugas-tugas pemrosesan data secara *real-time*. Dalam penelitian ini, Raspberry Pi 4 dipilih sebagai perangkat utama untuk menjalankan tugas ini karena spesifikasinya yang cukup mumpuni.

#### 2.4.1 Raspberry Pi

Raspberry Pi 4, dengan spesifikasi *Quad-core* ARM Cortex-A72, RAM 4 GB, *GPU Video Core VI* 500 Mhz, 4 *USB Port*, 2 *port micro-HDMI*, *Ethernet RJ45*, *Bluetooth 5.0*, serta *dual-band 2.4 GHz* dan *5 GHz WiFi*, mempertahankan bentuk yang serupa dengan model-model sebelumnya. Ini memungkinkan kesesuaian dengan sebagian besar aksesori dan perangkat keras tambahan yang telah ada, menawarkan fleksibilitas yang luas dalam berbagai aplikasi. Raspberry Pi 4 dapat digunakan mulai dari proyek hobi seperti komputer rumahan atau server media, hingga penggunaan profesional dalam pengembangan prototipe *IoT*, server kecil, dan sistem *embedded* lainnya. Didukung oleh berbagai distribusi Linux seperti Raspberry Pi OS dan Ubuntu, Raspberry Pi 4 juga menyediakan beragam proyek dan aplikasi yang dioptimalkan khusus untuk penggunaannya, memberikan kekuatan komputasi yang signifikan dalam sebuah platform yang kompak dan terjangkau (DATASHEET Raspberry Pi 4 Model B, 2024). Raspberry Pi 4 dapat dilihat pada Gambar 2.1.

Dalam konteks penelitian ini, Raspberry Pi 4 digunakan sebagai *edge sensor* untuk memonitor *access point*, mengumpulkan data, dan mengirimkan informasi tersebut ke dasbor yang telah dirancang. Fleksibilitas dan kemampuan komputasi dari Raspberry Pi 4 menjadikannya pilihan ideal untuk memfasilitasi pemantauan dan analisis jaringan, mendukung upaya untuk menjaga *robustness* dan *resilience* jaringan di UII.

Setelah membahas berbagai perangkat keras dan teknologi yang digunakan dalam penelitian ini, seperti *edge sensors* dan Raspberry Pi 4, penting untuk menyoroti peran perangkat lunak dalam mendukung keseluruhan sistem. Perangkat lunak ini tidak hanya memungkinkan pengumpulan dan analisis data, tetapi juga mendukung integrasi dan visualisasi hasil secara efisien.



Gambar 2.1 Raspberry Pi 4

## 2.5 Perangkat Lunak

Perangkat lunak atau peranti lunak adalah istilah yang mengacu pada data yang tersimpan dalam format digital, seperti program komputer, dokumentasi terkait, serta informasi lain yang dapat diproses oleh komputer. Ini adalah bagian dari sistem komputer yang tidak bersifat fisik dan berbeda dengan perangkat keras komputer. Studi ini memanfaatkan beberapa perangkat lunak pendukung dalam pelaksanaannya, yang berperan dalam mendukung proses penelitian yang dilakukan.

### 2.5.1 Docker

Docker adalah teknologi kontainerisasi yang memungkinkan pengembang untuk mengemas aplikasi beserta dependensinya ke dalam kontainer yang dapat dijalankan di berbagai lingkungan. Ini memastikan aplikasi berjalan konsisten di lingkungan pengembangan, pengujian, dan produksi, tanpa memedulikan perbedaan konfigurasi sistem (Potdar et al., 2020).

Dalam penelitian ini, Docker digunakan untuk memfasilitasi penyebaran dan pengelolaan layanan pemantauan, seperti Prometheus dan Grafana. Penggunaan Docker memudahkan dalam pengaturan lingkungan yang konsisten, meningkatkan isolasi antar aplikasi, dan mendukung skalabilitas serta kemudahan manajemen infrastruktur. Dengan

Docker, penelitian ini dapat memastikan bahwa sistem yang dibangun tetap *robust* dan mudah dikelola, sekaligus mengurangi potensi masalah yang disebabkan oleh perbedaan lingkungan.

### 2.5.2 Prometheus

Prometheus adalah sistem yang digunakan untuk pemantauan dan pemberitahuan dalam kerangka manajemen dan pemantauan. Sistem ini dirancang untuk mengumpulkan dan menganalisis data terkait pemantauan sistem komputasi skala besar. Prometheus membantu dalam menyediakan pemantauan waktu nyata dan visualisasi metrik sistem, berkontribusi pada pemantauan proaktif, skalabilitas, generasi peringatan, analisis tren, dan dasbor metrik praktis di lingkungan komputasi berkinerja tinggi (Sukhija & Bautista, 2019).

Prometheus dipilih dalam penelitian ini karena kemampuannya untuk menyediakan pemantauan waktu nyata dengan arsitektur yang efisien dan skalabilitas tinggi. Prometheus menggunakan metode *pull* untuk pengumpulan data yang memastikan data selalu terbaru, serta PromQL untuk analisis data yang mendalam. Dibandingkan dengan alat lain, Prometheus menawarkan fleksibilitas dalam integrasi dan visualisasi, seperti dengan Grafana, serta kemampuan untuk menangani volume data besar secara efektif. Keunggulan ini menjadikannya pilihan ideal untuk kebutuhan pemantauan sistem komputasi skala besar.

Untuk memaksimalkan manfaat dari data yang dikumpulkan oleh Prometheus, penting untuk memiliki alat yang dapat memvisualisasikan metrik tersebut dengan cara yang mudah dipahami dan diakses. Oleh karena itu, Grafana digunakan untuk memvisualisasikan data dari Prometheus.

### 2.5.3 Grafana

Grafana adalah sebuah platform *open-source* yang digunakan untuk visualisasi dan pemantauan data dalam bentuk grafik. Grafana merupakan alat yang sederhana dan cocok untuk memecahkan masalah terkait analisis data metrik. Grafana dapat digunakan untuk membuat dasbor yang memungkinkan pemantauan metrik kinerja secara *real-time*. Grafana juga dapat digunakan untuk membuat visualisasi hasil pengujian kinerja sistem, seperti histogram jumlah permintaan per detik, grafik jumlah permintaan per detik, persentil latensi permintaan, dan distribusi latensi permintaan. Dengan Grafana, pengguna dapat melacak dinamika hasil pengujian kinerja selama waktu (Ushakova et al., 2022).

Grafana dipilih karena kemampuannya dalam menyediakan visualisasi data yang intuitif dan interaktif. Sebagai alat yang terintegrasi dengan Prometheus, Grafana memungkinkan

pengguna untuk membuat dasbor yang kustomisasi dengan mudah, menyajikan data dalam format yang jelas dan dapat diakses. Fitur ini sangat penting untuk menganalisis data metrik yang dikumpulkan secara *real-time* dan membuat keputusan berbasis data yang cepat. Grafana juga mendukung berbagai jenis visualisasi yang membantu dalam memahami dan menginterpretasikan hasil pengujian kinerja, serta memungkinkan pelacakan tren secara efektif. Kemampuan Grafana untuk berintegrasi dengan alat lain dan menyajikan data secara dinamis menjadikannya pilihan ideal untuk memvisualisasikan dan memanfaatkan data yang dikumpulkan oleh Prometheus dalam penelitian ini.

Namun, karena *node-node* Prometheus berada pada Raspberry Pi 4 dan server Prometheus-Grafana berada di laptop yang berada di jaringan yang berbeda, diperlukan solusi untuk memastikan konektivitas yang aman dan andal antara kedua jaringan tersebut. Tailscale dapat digunakan untuk menjembatani koneksi antara Raspberry Pi 4 dan laptop dengan keamanan dan efisiensi.

#### 2.5.4 Tailscale

Tailscale Inc. adalah perusahaan perangkat lunak yang berbasis di Toronto, Ontario. Perusahaan ini mengembangkan perangkat lunak *mesh virtual private network* (VPN) yang sebagian bersifat *open-source* serta menyediakan layanan manajemen berbasis web. Tailscale menawarkan layanan VPN tanpa konfigurasi dengan nama yang sama.

Perangkat lunak *open-source* yang dikembangkan oleh Tailscale bekerja bersama dengan layanan manajemen untuk membangun komunikasi VPN *peer-to-peer* atau melalui relai dengan klien lain menggunakan protokol WireGuard. Jika perangkat lunak gagal untuk membangun komunikasi langsung, maka perangkat lunak tersebut akan menggunakan relai yang disediakan oleh perusahaan. Alamat IPv4 yang diberikan kepada klien berada dalam ruang yang disediakan untuk *carrier-grade* NAT, yang dipilih untuk menghindari gangguan dengan jaringan yang ada. Konfigurasi ini juga memungkinkan pengalihan lalu lintas ke jaringan di belakang klien pada beberapa klien, sehingga meningkatkan fleksibilitas dan fungsionalitas jaringan (Williams, 2024).

Tailscale dipilih karena kemudahan konfigurasi dan manajemen VPN yang ditawarkannya. Dengan Tailscale, pengguna dapat membangun jaringan VPN *mesh* tanpa perlu konfigurasi yang rumit, berkat integrasinya dengan protokol WireGuard yang menyediakan keamanan tinggi. Layanan ini memungkinkan komunikasi *peer-to-peer* dan relai yang fleksibel, serta menghindari masalah dengan jaringan yang ada melalui penggunaan alamat

IPv4 dalam ruang *carrier-grade* NAT. Tailscale juga mendukung pengalihan lalu lintas ke jaringan di belakang klien, meningkatkan fungsionalitas dan fleksibilitas jaringan. Keunggulan ini menjadikannya pilihan ideal untuk memastikan konektivitas yang aman dan efisien dalam sistem yang terintegrasi dengan Prometheus dan Grafana.

## 2.6 Penelitian Terkait

Terdapat 7 penelitian yang menjadi referensi dalam membuat perencanaan untuk penelitian ini. Penelitian terkait ini disajikan dalam bentuk tabel agar memudahkan dalam memahami tujuan penelitian dan fitur penting terkait.

Tabel 2.1 Penelitian Terdahulu

No.	Penulis	Tujuan Penelitian	Fitur Penting Terkait	Hasil Penelitian
1	(Kong & Yeh, 2009)	Menyelidiki ketahanan jaringan nirkabel terhadap kegagalan berjenjang menggunakan model berbasis perkolasi.  Menganalisis fenomena <i>cascading</i> dalam jaringan nirkabel skala besar dengan kendala geometris.	Makalah ini menggunakan beberapa metode untuk menganalisis ketahanan jaringan nirkabel terhadap kegagalan <i>node</i> , terutama dengan fokus pada kegagalan yang bergantung pada derajat dan kegagalan berantai.	Menganalisis kegagalan <i>cascading</i> dalam jaringan nirkabel skala besar menggunakan perspektif berbasis perkolasi.  Memperoleh kondisi analitis untuk kaskade dan transisi fase dalam model.
2	(Kakadia & Ramirez-Marquez, 2020)	Makalah ini berfokus pada pengoptimalan pengalaman pengguna jaringan melalui peningkatan ketahanan.	Menghubungkan pengalaman pengguna dengan ketahanan jaringan untuk	Kerangka kerja yang diusulkan meningkatkan pengalaman pengguna dan

No.	Penulis	Tujuan Penelitian	Fitur Penting Terkait	Hasil Penelitian
		<p>Kerangka kerja yang diusulkan mengotomatiskan identifikasi akar masalah untuk meningkatkan kinerja jaringan</p>	<p>pengoptimalan menggunakan pembelajaran mesin.</p> <p>Mengotomatiskan identifikasi gangguan jaringan dan akar penyebabnya untuk meningkatkan pengalaman pengguna.</p>	<p>ketahanan jaringan melalui otomatisasi.</p> <p>Prototipe mendeteksi anomali, mengidentifikasi akar penyebab, dan meningkatkan efisiensi jaringan.</p>
3	(Li et al., 2022)	<p>Strategi yang diusulkan meningkatkan ketahanan jaringan terhadap kegagalan yang bertingkat.</p> <p>Analisis teoretis dan eksperimen menunjukkan peningkatan efisiensi penguatan.</p>	<p>Mengusulkan strategi penguatan ketahanan untuk jaringan yang memiliki potensi risiko <i>cascade</i>.</p> <p>Mengembangkan metode penilaian ketahanan jaringan dengan mempertimbangkan struktur dan beban <i>nodal</i>.</p> <p>Meningkatkan strategi penguatan berdasarkan teori</p>	<p>Strategi ALCR mengungguli metode yang ada dalam hal efisiensi penguatan.</p> <p>Analisis teoretis dan eksperimen menunjukkan strategi ketahanan jaringan yang lebih baik.</p> <p>Nilai kritis dan fungsi penalti ditentukan untuk</p>

No.	Penulis	Tujuan Penelitian	Fitur Penting Terkait	Hasil Penelitian
			optimasi untuk ketahanan jaringan.	strategi penguatan.
4	(Zou et al., 2021)	<p>Model <i>Bizantium</i> yang diusulkan mengintegrasikan perilaku <i>jamming</i>, algoritma yang efisien dikembangkan.</p> <p>Algoritma menyelesaikan tugas penyebaran banyak pesan mendekati solusi optimal.</p> <p>Menjelaskan desain protokol yang toleran terhadap kesalahan untuk jaringan nirkabel yang sebenarnya.</p>	<p>Mengintegrasikan <i>jamming</i> ke dalam model <i>Bizantium</i> untuk efisiensi komunikasi jaringan nirkabel.</p> <p>Mengembangkan algoritma penyebaran beberapa pesan terdistribusi yang tahan terhadap pengaturan <i>byzantine</i> yang kuat.</p>	<p>Algoritma yang dikembangkan untuk penyebaran beberapa pesan dalam jaringan nirkabel dengan perilaku gangguan.</p> <p>Algoritma yang mendekati solusi optimal dalam hal waktu berjalan.</p> <p>Hasil empiris memvalidasi <i>byzantine-resilience</i> dan efisiensi algoritma.</p>
5	(Alpeni et al., 2019)	<p>Aplikasi Android memonitor parameter AP WiFi.Id untuk PT Telekomunikasi Indonesia.</p> <p>Menggunakan metode prototipe untuk</p>	<p>Pengembangan aplikasi Android untuk memonitor parameter WiFi.Id secara <i>real-time</i>.</p> <p>Memfasilitasi operator PT Telekomunikasi</p>	<p>Aplikasi Android untuk memonitor AP yang dikembangkan untuk PT Telekomunikasi Indonesia.</p>

No.	Penulis	Tujuan Penelitian	Fitur Penting Terkait	Hasil Penelitian
		<p>membangun aplikasi pemantauan</p> <p>Aplikasi yang memfasilitasi operator dalam memantau <i>Access Point</i> secara <i>real-time</i>.</p>	<p>Indonesia dalam mengelola <i>Access Point</i> secara efektif.</p>	<p>Aplikasi yang memudahkan operator dalam memonitoring <i>SSID</i>, <i>MAC Address</i>, dan lainnya.</p>
6	(Fu et al., 2021)	<p>Membahas masalah kegagalan bertingkat dalam jaringan sensor nirkabel (<i>WSN</i>), khususnya berfokus pada konfigurasi <i>multisink</i>. Ini memperkenalkan model <i>cascading</i> baru berdasarkan metrik "<i>multioriented link betweenness</i>" (<i>MLB</i>) dan mengusulkan algoritma <i>memetic</i> (<i>MA-MSP</i>) untuk mengoptimalkan penempatan <i>sink node</i> untuk meningkatkan ketahanan jaringan.</p>	<p>Mengusulkan model <i>cascading</i> yang realistis untuk <i>WSN multisink</i>.</p> <p>Mengembangkan algoritma evolusioner untuk mengoptimalkan penempatan <i>sink node</i>.</p>	<p>Model <i>cascading</i> yang diusulkan dievaluasi untuk <i>WSN multisink</i> menggunakan simulasi <i>MATLAB</i>.</p> <p><i>MA-MSP</i> meningkatkan ketahanan jaringan dan efisiensi komunikasi dalam <i>WSN multisink</i>.</p>

No.	Penulis	Tujuan Penelitian	Fitur Penting Terkait	Hasil Penelitian
7	(Pajic Miroslav, 2010)	<p>Seluruh jaringan bertindak sebagai pengontrol, menjaga stabilitas di bawah penurunan paket.</p> <p>Perilaku sistem dinamik linier dengan batasan <i>sparsity</i> yang diberlakukan oleh jaringan.</p> <p>Prosedur desain menstabilkan pabrik menggunakan kombinasi linier dari <i>node</i>.</p>	<p>Memperkenalkan konsep <i>Wireless Control Network (WCN)</i> untuk sistem kontrol jaringan.</p> <p>Mengembangkan prosedur desain numerik untuk menstabilkan pabrik dengan <i>WCN</i>.</p>	<p>Jaringan Kontrol Nirkabel yang diusulkan bertindak sebagai sistem dinamis linier.</p> <p>Prosedur desain menstabilkan pabrik dengan transmisi dari <i>node</i> yang paling dekat dengan aktuator.</p> <p>Algoritma memastikan stabilitas rata-rata kuadrat di bawah penurunan paket dalam jaringan.</p>

Dari tujuh penelitian yang diulas, berikut adalah kesimpulan dan identifikasi gap yang akan diisi oleh penelitian ini.

### 2.6.1 Kesimpulan Penelitian Terkait

- a. Kong & Yeh (2009) Menyelidiki ketahanan jaringan nirkabel terhadap kegagalan berjenjang dengan model berbasis perkolasi, memperoleh kondisi analitis untuk kaskade dan transisi fase dalam model.

- b. Kakadia & Ramirez-Marquez (2020) fokus pada peningkatan ketahanan jaringan dan pengalaman pengguna melalui otomatisasi identifikasi gangguan jaringan menggunakan pembelajaran mesin.
- c. Li et al. (2022) mengusulkan strategi penguatan ketahanan jaringan terhadap kegagalan berjenjang, menunjukkan peningkatan efisiensi penguatan melalui analisis teoretis dan eksperimen.
- d. Zou et al. (2021) mengembangkan model *Bizantium* yang mengintegrasikan perilaku *jamming*, menghasilkan algoritma penyebaran pesan yang efisien dan toleran terhadap gangguan.
- e. Alpeni et al. (2019) mengembangkan aplikasi Android untuk memonitor parameter AP WiFi.Id, memfasilitasi operator dalam mengelola Access Point secara *real-time*.
- f. Fu et al. (2021) memperkenalkan model *cascading* baru untuk WSN multisink dan mengusulkan algoritma evolusioner untuk mengoptimalkan penempatan *sink node*.
- g. Pajic Miroslav (2010) memperkenalkan konsep *Wireless Control Network (WCN)* untuk sistem kontrol jaringan, mengembangkan prosedur desain numerik untuk menstabilkan pabrik dengan *WCN*.

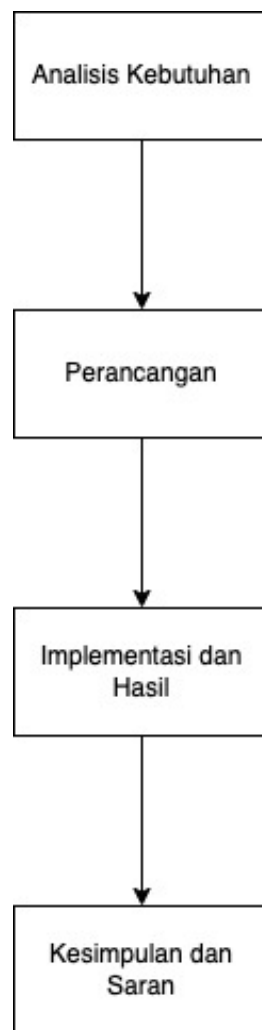
### 2.6.2 Gap Analysis

- a. Penggunaan Teknologi Terkini untuk Monitoring dan Pemberitahuan  
Tidak ada penelitian yang secara eksplisit menggabungkan teknologi modern seperti Prometheus dan Grafana untuk pemantauan jaringan secara *real-time* dan pemberitahuan otomatis. Penelitian ini akan mengisi gap dengan mengimplementasikan Prometheus dan Grafana untuk memantau dan memvisualisasikan kinerja jaringan, serta menggunakan Tailscale untuk memastikan konektivitas yang aman antara *node* Prometheus yang tersebar di jaringan yang berbeda.
- b. Integrasi Alat Pemantauan dan Sistem Pemberitahuan  
Tidak ada penelitian yang mengintegrasikan alat pemantauan dengan sistem pemberitahuan otomatis yang memberikan notifikasi dini kepada penanggung jawab jaringan. Penelitian ini akan mengisi gap dengan mengimplementasikan notifikasi melalui Telegram untuk memastikan respons cepat terhadap masalah jaringan.

## BAB III METODOLOGI PENELITIAN

### 3.1 Alur Pengerjaan Tugas Akhir

Proses pengerjaan tugas akhir ini dijelaskan melalui sebuah alur yang memberikan gambaran menyeluruh mengenai penelitian yang dilakukan, mulai dari tahap awal hingga penyelesaian akhir. Alur tersebut dapat dilihat pada Gambar 3.1



Gambar 3.1 Tahap pengerjaan

Berdasarkan Gambar 3.1, memiliki beberapa tahapan sebagai berikut:

a. Analisis Kebutuhan

Tujuan dari tahap ini adalah untuk mendapatkan pemahaman mendalam tentang cara terbaik untuk memantau jaringan dan memberitahu penanggung jawab mengenai *access*

*point* yang bermasalah. Data dikumpulkan melalui wawancara dengan narasumber atau calon pengguna aplikasi. Analisis ini akan membantu menentukan kebutuhan spesifik yang harus dipenuhi, termasuk fungsionalitas yang diperlukan dari Raspberry Pi 4.

b. Perancangan

Berdasarkan hasil analisis kebutuhan, tahap perancangan dilakukan. Tahap ini mencakup desain konfigurasi laptop dan Raspberry Pi 4 yang akan digunakan untuk memantau *access point*. Pada tahap ini, peneliti akan merancang parameter yang akan diukur dalam konteks jaringan, *node* yang digunakan, konfigurasi dasbor.

c. Implementasi dan Hasil

Pada tahap implementasi, konfigurasi awal Raspberry Pi 4 dan perancangan parameter yang akan diukur diterapkan dalam jaringan nyata. Raspberry Pi 4 ditempatkan di area UII yang memiliki jangkauan *access point* untuk memastikan perangkat mampu secara efektif mengumpulkan dan memantau data jaringan, serta mendeteksi dan menangani masalah *access point* dengan cepat. Penambahan konfigurasi variabel pada dasbor Grafana mempermudah pemilihan *node* yang akan dipantau, dengan visualisasi dalam bentuk *chart* untuk memudahkan pemantauan histori dan fluktuasi jaringan. Dari visualisasi ini, *threshold* dapat diterapkan untuk *alert* ketika *access point* mengalami anomali. Data yang dikumpulkan dari *speedtest*, *ping*, dan *traffic access point* kemudian dianalisis untuk menilai kinerja sistem. Setelah implementasi dan pengujian selesai, analisis hasil sistem pemantauan menjadi fokus utama.

d. Kesimpulan dan Saran

Evaluasi ini mencakup analisis kinerja jaringan menggunakan *edge sensor*, yaitu Raspberry Pi 4, selama jam kerja dan di luar jam kerja. Tahap ini menjelaskan bagaimana Raspberry Pi 4 berfungsi serta pengaruhnya dalam meningkatkan kualitas jaringan. Saran yang diberikan mencakup langkah-langkah prosedural untuk memastikan sistem pemantauan berjalan secara optimal dan efisien. Diharapkan penelitian ini dapat dikembangkan lebih lanjut di masa mendatang.

### 3.2 Analisis Kebutuhan

Analisis kebutuhan adalah proses untuk mendapatkan gambaran masalah yang muncul ketika tidak ada monitoring *access point*. Untuk mengumpulkan data yang dapat digunakan dalam analisis kebutuhan, dilakukan wawancara dengan narasumber atau calon pengguna aplikasi. Wawancara tidak terstruktur dianggap lebih efektif untuk menggali masalah dan

keinginan pengguna. Hasil wawancara kemudian disusun untuk memudahkan proses analisis kebutuhan.

### 3.2.1 Analisis Permasalahan

Analisis permasalahan merupakan tahap pertama dalam analisis kebutuhan. Tahap ini bertujuan untuk mengumpulkan gambaran masalah yang muncul ketika proses monitoring jaringan dilakukan dengan fokus pada gedung tertentu, bukan pada setiap *access point*. Akibat dari metode ini, data yang diperoleh sering kali tidak akurat dan tidak dapat mengidentifikasi *access point* yang bermasalah. Permasalahan utama ini menyebabkan proses monitoring jaringan menjadi tidak efisien dalam menangani *access point* yang bermasalah. Berdasarkan contoh di atas, berikut adalah beberapa poin permasalahan yang telah dirangkum:

- a. Ketika pengguna mengakses internet melalui *access point* UIIConnect atau eduroam dan mengalami masalah, mereka enggan melaporkannya kepada pihak Badan Sistem Informasi (BSI) untuk dilakukan pengecekan.
- b. Ketika pengguna UIIConnect atau eduroam melaporkan masalah pada jaringan *WiFi* ke BSI, sering kali setelah dilakukan pengecekan, tidak ditemukan masalah pada *access point*. Situasi ini menyebabkan kesulitan dalam mengidentifikasi dan mengatasi masalah secara efektif.

Berdasarkan beberapa poin permasalahan yang telah diidentifikasi, ditemukan bahwa Badan Sistem Informasi (BSI) belum memiliki visualisasi untuk memantau *access point*. Oleh karena itu, diperlukan implementasi proses monitoring *access point* yang disajikan dalam bentuk infografik.

### 3.2.2 Analisis Fitur Utama

Analisis fitur utama adalah proses untuk menggambarkan kebutuhan pengguna yang diidentifikasi melalui analisis permasalahan. Dari hasil analisis permasalahan, kebutuhan ini kemudian diubah menjadi fitur utama aplikasi. Hasil analisis permasalahan yang diterjemahkan menjadi fitur utama aplikasi dapat dilihat dalam Tabel 3.1.

Tabel 3.1 Permasalahan dan dasar fitur utama

Permasalahan	Dasar fitur utama
Ketika pengguna mengakses internet melalui <i>access point</i> UIIConnect atau eduroam dan	Sistem pemberitahuan untuk pencegahan dini ini dirancang agar pengguna tidak

mengalami masalah, mereka enggan melaporkannya kepada pihak Badan Sistem Informasi (BSI) untuk dilakukan pengecekan.	mengalami masalah saat menggunakan <i>access point</i> , serta memberikan peringatan kepada BSI untuk segera memperbaiki <i>access point</i> yang terdeteksi bermasalah.
Ketika pengguna UIIConnect atau eduroam melaporkan masalah pada jaringan <i>WiFi</i> ke BSI, sering kali setelah dilakukan pengecekan, tidak ditemukan masalah pada <i>access point</i> . Situasi ini menyebabkan kesulitan dalam mengidentifikasi dan mengatasi masalah secara efektif.	Visualisasi grafik untuk melihat riwayat kerja <i>access point</i> dan mengidentifikasi anomali yang dapat diinvestigasi lebih lanjut untuk menemukan penyebab masalah tersebut.

Fitur-fitur yang telah diidentifikasi ini menjadi landasan utama dalam perancangan sistem monitoring *access point* yang dirancang untuk memenuhi kebutuhan pengguna. Untuk memastikan bahwa fitur-fitur ini dapat diimplementasikan dengan efektif, proses analisis kebutuhan perangkat keras dan perangkat lunak menjadi sangat krusial. Proses ini memastikan bahwa setiap komponen yang dipilih tidak hanya mendukung fitur yang dirancang, tetapi juga bekerja secara optimal untuk menghasilkan sistem monitoring yang responsif dan andal.

### 3.2.3 Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Analisis kebutuhan perangkat keras dan perangkat lunak dilakukan melalui diskusi dengan Bapak Mukhammad Andri Setiawan, Kepala Badan Sistem Informasi (BSI) Universitas Islam Indonesia. Berdasarkan diskusi tersebut, peneliti mengidentifikasi kebutuhan perangkat keras serta beberapa perangkat lunak yang relevan untuk pengembangan sistem yang direncanakan:

a. Raspberry Pi 4

Raspberry Pi 4 digunakan sebagai alat untuk mengumpulkan informasi dari *access point*, termasuk data *speedtest*, *ping*, dan *traffic* antara *access point* dan Raspberry Pi 4. Perangkat ini berfungsi sebagai sensor jaringan yang mampu memantau kinerja dan kondisi *access point* secara *real-time*, sehingga mempermudah identifikasi dan penanganan masalah yang mungkin terjadi.

b. Prometheus

Prometheus digunakan dalam sistem ini sebagai alat untuk mengumpulkan dan menyimpan metrik dari berbagai sumber, termasuk Raspberry Pi 4. Prometheus dirancang untuk memantau dan mengawasi sistem, menyediakan fitur pengumpulan data yang efisien dan dapat disesuaikan. Dalam konteks ini, Prometheus akan mengumpulkan data metrik seperti *speedtest*, *ping*, dan *traffic* dari *access point* yang diambil oleh Raspberry Pi 4. Data ini kemudian akan disimpan dan dapat diakses untuk analisis lebih lanjut, membantu dalam identifikasi anomali dan permasalahan jaringan secara lebih efektif.

c. Grafana digunakan dalam sistem ini sebagai alat visualisasi untuk menampilkan data yang dikumpulkan oleh Prometheus. Grafana memungkinkan pembuatan dasbor interaktif yang memudahkan pengguna dalam memantau kinerja jaringan secara *real-time*. Dengan menggunakan Grafana, data metrik seperti hasil *speedtest*, *ping*, dan *traffic* dari *access point* yang dikumpulkan oleh Raspberry Pi 4 dan disimpan oleh Prometheus dapat divisualisasikan dalam bentuk grafik yang informatif.

d. Docker

Docker digunakan dalam sistem ini untuk menjalankan aplikasi monitoring dalam lingkungan yang terisolasi dan konsisten. Docker akan diinstal pada laptop terpisah, bukan pada Raspberry Pi 4. Dengan Docker, berbagai aplikasi yang diperlukan untuk monitoring, seperti Prometheus dan Grafana, dapat di-*deploy* sebagai kontainer. Hal ini memungkinkan instalasi dan konfigurasi yang lebih mudah serta memastikan bahwa aplikasi berjalan dengan stabil tanpa tergantung pada sistem operasi laptop.

Berdasarkan analisis kebutuhan, setiap komponen yang dipilih memiliki peran penting dalam perancangan dan pengembangan aplikasi, yang dirancang untuk memenuhi kebutuhan pengguna secara optimal dan efektif. Raspberry Pi 4 berfungsi sebagai sensor jaringan utama, mengumpulkan informasi dari *access point* seperti data *speedtest*, *ping*, dan *traffic*, serta memungkinkan pemantauan kinerja dan kondisi *access point* secara *real-time*. Prometheus

digunakan untuk mengumpulkan dan menyimpan metrik dari berbagai sumber, termasuk Raspberry Pi 4, dengan menawarkan fitur pengumpulan data yang efisien dan dapat disesuaikan. Data yang dikumpulkan, seperti *speedtest*, *ping*, dan *traffic*, disimpan oleh Prometheus untuk analisis lebih lanjut dan membantu dalam identifikasi anomali jaringan. Grafana berperan dalam visualisasi data dengan menyediakan dasbor interaktif yang menyajikan data secara grafis, memudahkan pemantauan kinerja jaringan secara *real-time* dan memungkinkan analisis yang mendalam. Docker, yang diinstal pada laptop terpisah dari Raspberry Pi 4, digunakan untuk menjalankan aplikasi monitoring seperti Prometheus dan Grafana dalam lingkungan yang terisolasi dan konsisten, memastikan konfigurasi dan instalasi yang lebih sederhana serta kestabilan aplikasi tanpa bergantung pada sistem operasi laptop. Pemilihan kombinasi perangkat keras dan perangkat lunak ini dilakukan karena kemampuan mereka untuk menyediakan solusi monitoring yang komprehensif, fleksibel, dan efisien, sesuai dengan kebutuhan sistem yang direncanakan.

### 3.3 Perancangan

Perancangan merupakan fase yang bertujuan mewujudkan ide awal dalam pengembangan aplikasi. Ide awal ini mencakup langkah-langkah untuk menyelesaikan atau mengembangkan aplikasi, yang didasarkan pada hasil analisis kebutuhan sebelumnya. Langkah-langkah tersebut meliputi instalasi *node* pada Raspberry Pi 4, instalasi Prometheus dan Grafana pada menggunakan Docker di Windows, integrasi Raspberry Pi 4 dengan Prometheus pada Docker di Windows, penyajian data monitoring *access point* pada Grafana melalui Prometheus, pengaturan dasbor, konfigurasi Tailscale dan *UFW firewall*.

#### 3.3.1 Instalasi *Node* Pada Raspberry Pi 4

Instalasi *node* dilakukan setelah Raspberry Pi 4 siap digunakan, dengan Raspberry Pi OS sudah terpasang. *Node* yang diinstal disesuaikan dengan kebutuhan utama, yaitu memantau kecepatan unduh, kecepatan unggah, *jitter*, *ping*, dan *traffic* antara *access point* dan Raspberry Pi 4. *Node* yang digunakan adalah *Node Exporter* (Ziemke, 2020), *Ping Exporter* (Brendgen, 2024), dan *Speedtest Exporter* (Carvalho, 2023). Ketiga *node* ini mencakup semua kebutuhan yang telah dijelaskan sebelumnya.

*Node* ini diatur untuk memulai kembali *service node* nya jika Raspberry Pi 4 memulai ulang perangkatnya, dengan membuat pengaturan *systemd* unit fail.

a. *Node Exporter*

```
[Unit]
Description=Node Exporter

[Service]
# Provide a text file location for
https://github.com/fahlke/raspberrypi_exporter data with the
#--collector.textfile.directoryparameter.
ExecStart=/usr/local/bin/node_exporter--collector.textfile.directory
/var/lib/node_exporter/textfile_collector

[Install]
WantedBy=multi-user.target
```

Gambar 3.2 Instalasi *Node Exporter*

Gambar 3.2 adalah konfigurasi *systemd* untuk menjalankan *Node Exporter* sebagai layanan di sistem operasi berbasis Linux. Bagian *[Unit]* memberikan deskripsi singkat tentang layanan ini. Bagian *[Service]* menentukan perintah yang akan dijalankan (*ExecStart*) untuk memulai *Node Exporter*, termasuk parameter *--collector.textfile.directory* yang menunjukkan lokasi fail teks untuk data yang dikumpulkan. Bagian *[Install]* mengindikasikan bahwa layanan ini harus dimulai dalam mode *multi-user*. Setelah mengonfigurasi fail layanan ini, perintah berikut digunakan untuk mengelola layanan *Node Exporter*:

```
sudo systemctl daemon-reload
sudo systemctl enable node_exporter.service
sudo systemctl start node_exporter.service
```

Gambar 3.3 *Prompt* mengulang kembali sistem

Penjelasan dari Gambar 3.3 untuk prompt tersebut

12. '*sudo systemctl daemon-reload*': Memuat ulang konfigurasi *systemd* untuk memastikan perubahan diterapkan.
13. '*sudo systemctl enable node\_exporter.service*': Mengaktifkan layanan *Node Exporter* agar otomatis berjalan saat sistem *boot*.
14. '*sudo systemctl start node\_exporter.service*': Memulai layanan *Node Exporter* segera.

## b. Ping Exporter

```
[Unit]
Description=Ping Exporter
After=network.target

[Service]
User=ping_exporter
ExecStart=/usr/bin/ping_exporter --
config.path=/etc/ping_exporter/config.yml

# This unit assumes systemd 232, present in EdgeOS 2.0.0
# (a derivative of Vyatta/Debian 9).
#
# If the ping_exporter was installed on system with a newer systemd
# version, you'll find additional drop-ins in ping_exporter.d/.

CapabilityBoundingSet=CAP_NET_RAW
AmbientCapabilities=CAP_NET_RAW
PrivateDevices=true
PrivateTmp=yes
ProtectControlGroups=true
ProtectKernelModules=yes
ProtectKernelTunables=true
ProtectSystem=strict
ProtectHome=true
DevicePolicy=closed
RestrictRealtime=yes
MemoryDenyWriteExecute=yes

[Install]
WantedBy=default.target
```

Gambar 3.4 Instalasi *Ping Exporter*

Gambar 3.4 di atas adalah konfigurasi *systemd* untuk menjalankan *Ping Exporter* sebagai layanan di sistem operasi berbasis Linux. Berikut adalah penjelasan tiap bagian:

### 1. [Unit]:

- ‘*Description=Ping Exporter*’: Memberikan deskripsi singkat tentang layanan ini.
- ‘*After=network.target*’: Menentukan bahwa layanan ini harus dimulai setelah jaringan aktif.

### 2. [Service]:

- ‘*User=ping\_exporter*’: Menentukan pengguna yang akan menjalankan layanan ini.
- ‘*ExecStart=/usr/bin/ping\_exporter --config.path=/etc/ping\_exporter/config.yml*’: Menentukan perintah untuk memulai *Ping Exporter* dengan jalur konfigurasi yang ditentukan.
- Keamanan dan Pembatas
- ‘*CapabilityBoundingSet=CAP\_NET\_RAW*’: Membatasi kemampuan yang dimiliki oleh layanan ini.

- '*AmbientCapabilities=CAP\_NET\_RAW*': Menambahkan kemampuan untuk menggunakan *CAP\_NET\_RAW*.
- '*PrivateDevices=true*': Membuat perangkat menjadi privat untuk layanan ini.
- '*PrivateTmp=yes*': Membuat direktori */tmp* menjadi privat untuk layanan ini.
- '*ProtectControlGroups=true*': Melindungi grup kontrol dari perubahan.
- '*ProtectKernelModules=yes*': Melindungi modul *kernel* dari modifikasi.
- '*ProtectKernelTunables=true*': Melindungi parameter *kernel* dari modifikasi.
- '*ProtectSystem=strict*': Membatasi akses layanan ke sistem fail.
- '*ProtectHome=true*': Melindungi direktori *home* dari akses layanan.
- '*DevicePolicy=closed*': Membatasi akses layanan ke perangkat.
- '*RestrictRealtime=yes*': Membatasi penggunaan waktu nyata oleh layanan.
- '*MemoryDenyWriteExecute=yes*': Mencegah eksekusi dari memori yang dapat ditulis.

### 3. [Install]

- *WantedBy=default.target*: Menentukan bahwa layanan ini akan dimulai dalam target *default*.

Penjelasan ini memberikan gambaran umum tentang bagaimana *Ping Exporter* diatur dan dijalankan sebagai layanan *systemd*, dengan berbagai langkah keamanan yang diterapkan untuk membatasi akses dan kemampuan layanan tersebut pada sistem.

```
sudo systemctl daemon-reload
sudo systemctl enable ping_exporter.service
sudo systemctl start ping_exporter.service
```

Gambar 3.5 *Prompt* mengulang kembali sistem

Penjelasan dari Gambar 3.5 untuk *prompt* tersebut

- '*sudo systemctl daemon-reload*': Memuat ulang konfigurasi *systemd* untuk memastikan perubahan diterapkan.
- '*sudo systemctl enable ping\_exporter.service*': Mengaktifkan layanan *Ping Exporter* agar otomatis berjalan saat sistem *boot*.
- '*sudo systemctl start ping\_exporter.service*': Memulai layanan *Ping Exporter* segera.

### c. *Speedtest Exporter*

```
[Unit]
Description=Monitor Kecepatan Internet

[Service]
User=ul
WorkingDirectory=/home/ul/speedtest_exporter/speedtest-exporter
ExecStart=/bin/bash -c 'cd /home/ul/speedtest_exporter/speedtest-exporter &&
source .venv/bin/activate && python src/exporter.py'
# optional items below
Restart=always
RestartSec=3

[Install]
WantedBy=multi-user.target
```

Gambar 3.6 Instalasi *Speedtest Exporter*

Penjelasan dari Gambar 3.6 untuk perintah-perintah dalam fail unit *systemd* ini:

1. *[Unit]*
  - *'Description=Monitor Kecepatan Internet'*: Menyediakan deskripsi singkat mengenai layanan ini, yaitu untuk memonitor kecepatan internet.
2. *[Service]*
  - *'User=ul'*: Menentukan pengguna yang akan menjalankan layanan ini, yaitu 'ul'.
  - *'WorkingDirectory=/home/ul/speedtest\_exporter/speedtest-exporter'*: Menentukan direktori kerja untuk layanan ini.
  - *'ExecStart=/bin/bash -c 'cd /home/ul/speedtest\_exporter/speedtest-exporter && source .venv/bin/activate && python src/exporter.py''*: Perintah untuk memulai layanan. Perintah ini melakukan:
    - Berpindah ke direktori kerja */home/ul/speedtest\_exporter/speedtest-exporter*.
    - Mengaktifkan lingkungan virtual *Python*.
    - Menjalankan skrip *Python* *'exporter.py'* untuk memulai proses ekspor data kecepatan internet.
  - *'Restart=always'*: Mengatur layanan untuk selalu *restart* jika terjadi kegagalan.
  - *'RestartSec=3'*: Menentukan waktu tunggu selama 3 detik sebelum mencoba *restart* layanan setelah kegagalan.
3. *[Install]*
  - *WantedBy=multi-user.target*: Menentukan target instalasi layanan ini untuk memastikan layanan berjalan pada mode *multi-user*.

Penjelasan ini memberikan gambaran umum tentang bagaimana *Speedtest Exporter* diatur dan dijalankan sebagai layanan *systemd*, dengan berbagai langkah keamanan yang diterapkan untuk membatasi akses dan kemampuan layanan tersebut pada sistem.

```
sudo systemctl daemon-reload
sudo systemctl enable speedtest_exporter.service
sudo systemctl start speedtest_exporter.service
```

Gambar 3.7 *Prompt* mengulang kembali sistem

Penjelasan dari Gambar 3.7 untuk *prompt* tersebut

1. '*sudo systemctl daemon-reload*': Memuat ulang konfigurasi *systemd* untuk memastikan perubahan diterapkan.
2. '*sudo systemctl enable speedtest\_exporter.service*': Mengaktifkan layanan *Speedtest Exporter* agar otomatis berjalan saat sistem *boot*.
3. '*sudo systemctl start speedtest\_exporter.service*': Memulai layanan *Speedtest Exporter* segera.

### 3.3.2 Instalasi Prometheus dan Grafana menggunakan Docker di Windows

Instalasi Prometheus dan Grafana dilakukan menggunakan Docker untuk memastikan kemudahan pengelolaan dan konsistensi lingkungan. Docker adalah platform *open-source* yang memungkinkan pengguna untuk mengotomatisasi pengiriman, penskalaan, dan pengelolaan aplikasi dalam wadah terisolasi. Prometheus adalah sistem monitoring dan alerting *open-source* yang dirancang untuk merekam metrik secara *real-time* dalam basis data deret waktu, sementara Grafana adalah platform analitik *open-source* yang menyediakan alat visualisasi yang kuat. Dengan menggunakan Docker, instalasi kedua alat ini dapat dilakukan dengan lebih efisien dan terstandarisasi, sehingga meminimalisir masalah kompatibilitas dan mempercepat proses pengembangan.

#### a. Persiapan Environment

Sebelum memulai instalasi Prometheus dan Grafana, pastikan Docker Desktop sudah terpasang dan berjalan dengan baik di sistem Windows yang digunakan. Ikuti langkah-langkah berikut untuk menginstal Docker Desktop:

1. Unduh Docker Desktop: Kunjungi situs resmi Docker dan unduh Docker Desktop untuk Windows.

2. Instal Docker Desktop: Jalankan fail installer yang telah diunduh dan ikuti petunjuk instalasi. Pastikan untuk mengaktifkan WSL 2 (Windows Subsystem for Linux 2) selama instalasi.
3. Jalankan Docker Desktop: Setelah instalasi selesai, jalankan Docker Desktop. Docker akan membutuhkan beberapa saat untuk memulai. Pastikan ikon Docker muncul di system tray dan statusnya "Running".

- Pull Image Docker

Langkah pertama adalah pull image Docker untuk Prometheus dan Grafana dari Docker Hub. Buka *Command Prompt* atau *PowerShell*, lalu jalankan perintah seperti Gambar 3.8 berikut:

```
docker pull prom/prometheus
docker pull grafana/grafana
```

Gambar 3.8 *Prompt* mengambil *packet* Prometheus dan Grafana

- Menjalankan Kontainer Prometheus

Setelah *image* berhasil di-*pull*, jalankan kontainer Prometheus dengan perintah seperti Gambar 3.9 berikut:

```
docker run -d --name=prometheus -p 9090:9090 prom/prometheus
("Monitoring Jenkins with Grafana and Prometheus - GitHub")
```

Gambar 3.9 *Prompt* menjalankan Prometheus

Perintah ini akan menjalankan Prometheus di latar belakang (-d) dan memetakan *port* 9090 pada hos ke *port* 9090 di dalam kontainer.

- Menjalankan Kontainer Grafana

Selanjutnya, jalankan kontainer Grafana dengan perintah seperti Gambar 3.10 berikut:

```
"docker run -d --name=grafana -p 3000:3000 grafana/grafana" ("How o
install & configure the grafana on AWS ec2")
```

Gambar 3.10 *Prompt* menjalankan Grafana

Perintah ini akan menjalankan Grafana di latar belakang dan memetakan *port* 3000 pada *hos* ke *port* 3000 di dalam kontainer.

– Mengkonfigurasi Prometheus

Setelah kontainer berjalan, konfigurasi Prometheus perlu disesuaikan untuk mulai memantau metrik. Buat fail konfigurasi `prometheus.yml` seperti Gambar 3.11 berikut:

```
global:
  scrape_interval: 15s

scrape_configs:
  - job_name: 'prometheus'
    static_configs:
      - targets: ['localhost:9090']
```

Gambar 3.11 *Config* prometheus.yml

Simpan fail ini di direktori yang dapat diakses oleh Docker, misalnya di `A:\Docker\prometheus.yml`. Kemudian jalankan kembali kontainer Prometheus dengan mengaitkan fail konfigurasi seperti Gambar 3.12 berikut:

```
docker run -d --name=prometheus -p 9090:9090 -v
A:\docker\prometheus.yml:/etc/prometheus/prometheus.yml prom/Prometheus
```

Gambar 3.12 *Prompt* menjalankan prometheus.yml pada Prometheus

– Mengakses Grafana dan Menambahkan Data *Source* Prometheus

Setelah kedua kontainer berjalan, buka browser dan akses Grafana melalui `http://localhost:3000`. *Login* dengan *username* dan *password default* (admin/grafana), kemudian tambahkan Prometheus sebagai data source:

- Pilih "Configuration" -> "Data Sources" -> "Add data source".
- Pilih "Prometheus" dan isi URL dengan `http://prometheus:9090/`.
- Klik "Save & Test" untuk memastikan Grafana dapat terhubung ke Prometheus.

### 3.3.3 Integrasi Raspberry Pi 4 dengan Prometheus pada Docker di Windows

Isi subbab ini menjelaskan langkah-langkah untuk menghubungkan Raspberry Pi 4 yang telah diinstal dengan *node-node* monitoring (*node\_exporter*, *ping\_exporter*, *speedtest\_exporter*) ke Prometheus yang diinstal pada Docker di Windows.

#### a. Prometheus Instalasi *Node* pada Raspberry Pi 4

Pastikan Raspberry Pi 4 telah diinstal dengan Raspberry Pi OS dan *node-node* yang diperlukan:

1. Node Exporter
2. Ping Exporter
3. Speedtest Exporter

Konfigurasi setiap *node* agar dapat mengumpulkan data yang diperlukan.

#### b. Mongonfigurasi Prometheus pada Docker di Windows

1. Jalankan kontainer Prometheus menggunakan Docker dengan perintah seperti Gambar 3.13:

```
docker run -d --name prometheus -p 9090:9090 prom/Prometheus
```

Gambar 3.13 Prompt menjalankan Prometheus di Docker

2. Buat atau modifikasi fail konfigurasi *prometheus.yml* pada Docker untuk menambahkan target *node-node* dari Raspberry Pi 4. Konfigurasinya dapat dilihat pada Gambar 3.14.

```
global:
  scrape_interval:           1m # By default, scrape targets every 15
    seconds.
  external_labels:
    monitor: 'codelab-monitor'

scrape_configs:
  - job_name: 'prometheus'
    # Override the global default and scrape targets from this job
    every 5 seconds.
    scrape_interval: 5s
    static_configs:
      - targets: ['prometheus:9090']

      - job_name: 'pi4 node_exporter'
```

```

scrape_interval: 5s
static_configs:
  - targets: ['100.125.16.50:9100']

- job_name: 'pi4 speedtest_exporter'
  scrape_interval: 180m
  scrape_timeout: 3m
  static_configs:
    - targets: ['100.125.16.50:9798']

- job_name: 'pi4 ping_exporter'
  scrape_interval: 5s
  scrape_timeout: 3s
  static_configs:
    - targets: ['100.125.16.50:9427']

```

Gambar 3.14 Prompt menjalankan *node*

### 3. Menghubungkan Raspberry Pi 4 ke Prometheus

- Pastikan *node-node* pada Raspberry Pi 4 berjalan dan dapat diakses melalui jaringan.
- Konfigurasi *firewall* atau aturan jaringan lain agar Prometheus di Windows dapat mengakses Raspberry Pi 4.
- Verifikasi koneksi dengan membuka Prometheus *web interface* di `'http://localhost:9090/targets'` dan memastikan semua target *node* terlihat dan statusnya UP.

#### 3.3.4 Penyajian Data Monitoring Access Point pada Grafana melalui Prometheus

Pada subbab ini, akan dijelaskan proses visualisasi data monitoring yang dikumpulkan dari Raspberry Pi 4 menggunakan Grafana. Grafana adalah alat analitik *open-source* yang memungkinkan pembuatan, eksplorasi, dan berbagi dasbor. Dalam konteks ini, data dari *Node Exporter*, *Ping Exporter*, dan *Speedtest Exporter* akan divisualisasikan melalui Grafana.

##### a. Mengimpor Dasbor di Grafana

1. Node Exporter Dashboard
2. Impor dasbor khusus untuk Node Exporter dengan menggunakan ID dasbor dari Grafana Dashboard Repository. Gunakan ID 1860 untuk "Node Exporter Full".
3. Ping Exporter Dashboard

4. Impor dasbor khusus untuk Ping Exporter dengan menggunakan ID dasbor dari Grafana Dashboard Repository. Gunakan ID 19761 untuk "Ping Exporter".
5. Speedtest Exporter Dashboard
6. Impor dasbor khusus untuk Speedtest Exporter dengan menggunakan ID dasbor dari Grafana Dashboard Repository. Gunakan ID 12473 untuk "Speedtest Exporter".

b. Menguji Dasbor

Setelah ketiga dasbor diimpor, pastikan untuk menguji masing-masing dasbor dengan data yang dikumpulkan oleh *node-node* yang relevan. Verifikasi bahwa data untuk kecepatan unduh, kecepatan unggah, *jitter*, *ping*, dan *traffic* terlihat dengan benar dan sesuai.

c. Menggabungkan Data di Node Exporter Dashboard

Setelah memastikan semua dasbor berjalan dengan baik, langkah selanjutnya adalah menggabungkan data yang diperlukan ke dalam dasbor utama, yaitu *Node Exporter Dashboard*. Ambil beberapa *query* dari *Ping Exporter Dashboard* dan *Speedtest Exporter Dashboard* sesuai dengan kebutuhan, lalu masukkan ke dalam *Node Exporter Dashboard*. Query untuk Ping Exporter dapat dilihat pada Gambar 3.15:

```
#untuk ping 8.8.8.8 dengan IPv4 query best
avg by(ip_version) (ping_rtt_best_seconds{job=~"pi4 ping_exporter",
instance=~"100.125.16.50:9427",           target=~"8\\.8\\.8\\.8",
ip_version=~"(4|6)"})

#untuk ping google.com dengan IPv4 dan IPv6 query best
avg by(ip_version) (ping_rtt_best_seconds{job=~"pi4 ping_exporter",
instance=~"100.125.16.50:9427",           target=~"google\\.com",
ip_version=~"(4|6)"})
```

Gambar 3.15 *Query ping* 8.8.8.8 dan google.com pada Grafana

*Query* pertama digunakan untuk menghitung rata-rata waktu respons terbaik (*ping RTT*) ke tujuan 8.8.8.8 menggunakan IPv4. *Query* ini menggunakan ekspresi `avg by(ip_version)` untuk mengelompokkan hasil berdasarkan versi IP (IPv4 atau IPv6). Data yang digunakan berasal dari `ping_rtt_best_seconds` yang dipetakan dengan pekerjaan `pi4 ping_exporter`, di mana instance yang terhubung adalah `100.125.16.50:9427`. Target dari ping adalah `8.8.8.8`, dengan mengizinkan pilihan untuk versi IP yang dapat diterima (`ip_version=~"(4|6)"`).

*Query* kedua mirip dengan yang pertama namun menargetkan `google.com` untuk *ping*, memungkinkan untuk penghitungan rata-rata waktu respons terbaik (*ping RTT*) dengan mendukung baik IPv4 maupun IPv6. Ini juga menggunakan ekspresi `avg by(ip\_version)` untuk memisahkan hasil berdasarkan versi IP, dan menggunakan data dari `ping\_rtt\_best\_seconds` yang tercatat oleh `pi4 ping\_exporter`, dengan *instance* yang sama dan `100.125.16.50:9427` sebagai *endpointnya*.

Penggunaan 8.8.8.8 sebagai target *ping* dalam *query* ini dipilih karena IP tersebut adalah salah satu alamat server *DNS* publik yang disediakan oleh Google, yang dikenal dengan kehandalannya dan distribusi global yang luas. 8.8.8.8 sering digunakan sebagai tolok ukur umum untuk pengujian jaringan karena jaringan Google memiliki infrastruktur yang stabil dan tersedia di berbagai lokasi, sehingga memberikan hasil yang konsisten untuk mengukur latensi dan kinerja konektivitas internet. Meskipun alamat lain seperti 1.1.1.1 (Cloudflare) atau 9.9.9.9 (Quad9) juga dapat digunakan untuk tujuan yang sama, 8.8.8.8 tetap menjadi pilihan yang lebih populer di kalangan pengguna dan profesional IT karena sudah lama menjadi standar *de facto* untuk pengujian jaringan, sehingga memudahkan perbandingan hasil dengan metrik yang sudah ada.

### 3.3.5 Pengaturan Dasbor dan Penentuan *Baseline*

Tahap ini bertujuan untuk menyempurnakan dasbor Grafana agar dapat menampilkan data dengan jelas dan menentukan *baseline* jaringan. Penyesuaian ini mencakup pengaturan visualisasi data menggunakan *query* yang tepat dan penentuan batasan performa jaringan yang akan digunakan sebagai acuan standar dalam monitoring.

a. Pada tahap ini, penyesuaian dasbor di Grafana dilakukan untuk memastikan bahwa semua informasi penting terlihat jelas dan mudah dipahami. Proses ini mencakup penambahan dan pengaturan panel, grafik, serta visualisasi lainnya sesuai kebutuhan, dengan tujuan memberikan gambaran komprehensif tentang kinerja *access point* dan jaringan.

#### 1. Menambahkan Panel

Pertama, tambahkan panel baru ke dasbor Grafana dengan mengklik tombol "*Add Panel*". Pilih jenis grafik atau visualisasi yang sesuai, seperti garis, batang, atau *gauge*, untuk menampilkan data yang diinginkan. Setiap panel harus diatur untuk menampilkan data spesifik seperti *ping*, *speedtest*, atau informasi lain yang relevan.

#### 2. Menggunakan *Query* Grafana

Gunakan *query* Grafana untuk menarik data ke dalam panel. Contohnya, untuk menampilkan *ping* ke 8.8.8.8, gunakan *query* pada Gambar 3.16 berikut:

```
avg by(ip_version) (ping_rtt_best_seconds{job=~"pi4 ping_exporter",
instance=~"100.125.16.50:9427", target=~"8\\.8\\.8\\.8",
ip_version=~"(4|6)"}) * 1000
```

Gambar 3.16 *Query ping* 8.8.8.8 pada Grafana

*Query* ini akan menampilkan nilai rata-rata *ping* ke 8.8.8.8 yang diambil oleh Raspberry Pi 4, yang diformat dalam milidetik.

### 3. Mengatur Tata Letak

Setelah panel ditambahkan, atur tata letak dasbor agar mudah dibaca dan dipahami. Tambahkan judul dan deskripsi pada setiap panel untuk memberikan konteks yang jelas kepada pengguna.

#### b. Penentuan Baseline Jaringan

Setelah dasbor disesuaikan, langkah berikutnya adalah menentukan baseline jaringan. Baseline ini berfungsi sebagai acuan standar untuk menilai performa jaringan dan mengidentifikasi anomali. Proses ini dimulai dengan menganalisis data historis yang telah dikumpulkan selama periode tertentu, seperti data ping, speedtest, dan traffic dari access point.

##### 1. Analisis Data Historis

Untuk menetapkan baseline, data historis dianalisis guna menentukan nilai rata-rata dan standar deviasi untuk setiap metrik yang dipantau. Misalnya, jika ping rata-rata ke server tertentu adalah 20 *ms* dengan standar deviasi 5 *ms*, maka nilai yang melebihi 25 *ms* atau kurang dari 15 *ms* bisa dianggap sebagai anomali.

##### 2. Mengintegrasikan Baseline ke Dasbor

Setelah baseline ditetapkan, masukkan nilai-nilai ini ke dalam Grafana sebagai threshold pada setiap panel. Threshold ini akan memicu alarm jika data yang masuk menyimpang dari baseline yang ditentukan, memungkinkan tim IT untuk segera mengenali dan menanggapi masalah yang muncul.

### 3.3.6 Konfigurasi Tailscale dan UFW Firewall

Pada subbab ini, akan dijelaskan proses konfigurasi Tailscale dan UFW Firewall pada Raspberry Pi 4 untuk memastikan konektivitas yang aman antara Raspberry Pi 4 dan perangkat

Windows. Langkah-langkah ini memastikan bahwa akses ke Raspberry Pi 4 hanya dapat dilakukan melalui jaringan Tailscale, memberikan lapisan keamanan tambahan.

#### a. Instalasi dan Konfigurasi Tailscale

Tailscale adalah alat VPN yang memungkinkan koneksi antara perangkat secara aman dan mudah. Langkah-langkah berikut akan menunjukkan cara menginstal dan mengkonfigurasi Tailscale pada Raspberry Pi 4 dan perangkat Windows.

##### 1. Instalasi Tailscale pada Raspberry Pi 4

Untuk menginstal Tailscale pada Raspberry Pi 4, buka terminal dan jalankan perintah berikut: ``curl -fsSL https://tailscale.com/install.sh | sh``. Setelah instalasi selesai, aktifkan Tailscale dengan perintah ``sudo tailscale up``. Ikuti instruksi yang muncul untuk masuk ke akun Tailscale dan tambahkan perangkat Raspberry Pi ke jaringan Tailscale.

##### 2. Instalasi Tailscale pada Perangkat Windows

Untuk menginstal Tailscale pada perangkat Windows, unduh *installer* dari situs resmi Tailscale. Setelah mengunduh, jalankan *installer* dan ikuti petunjuk instalasi yang muncul. Setelah selesai menginstal, buka aplikasi Tailscale dan masuk menggunakan akun untuk menambahkan perangkat Windows ke jaringan Tailscale.

#### b. Penggunaan Tailscale untuk Akses Aman

Dengan Tailscale dan *UFW Firewall* yang dikonfigurasi, akses ke Raspberry Pi 4 dari perangkat Windows hanya dapat dilakukan melalui jaringan Tailscale, memastikan koneksi yang aman.

### 3.3.7 Konfigurasi Dasbor Grafana

Sub bab ini menjelaskan langkah-langkah konfigurasi dasbor Grafana untuk mendukung pemantauan kinerja *access point* dan jaringan secara *real-time*. Proses yang dijelaskan meliputi instalasi dan konfigurasi Prometheus, penghubungan Raspberry Pi 4 sebagai *node* pemantauan, serta impor dan penyesuaian dasbor di Grafana untuk visualisasi data metrik secara komprehensif. Penyesuaian ini bertujuan untuk memberikan informasi yang mendetail mengenai kesehatan jaringan dan memudahkan analisis performa jaringan secara efektif.

#### a. Instalasi dan Konfigurasi Prometheus

Pada tahap ini, Prometheus diinstal dan dikonfigurasi pada Docker yang berjalan di sistem operasi Windows. Prometheus digunakan untuk mengelola dan menyimpan data metrik yang dikumpulkan dari berbagai *node*. Dengan menggunakan Docker, instalasi dan

manajemen Prometheus menjadi lebih fleksibel dan terisolasi dari sistem hos, sehingga memudahkan pengaturan dan pemeliharaan.

b. Menghubungkan Raspberry Pi 4 ke Prometheus

Raspberry Pi 4 dikonfigurasi untuk mengirimkan data metrik yang dikumpulkan ke Prometheus yang berjalan di Docker. Konfigurasi ini memastikan bahwa semua metrik dari *node-node* seperti *node\_exporter*, *ping\_exporter*, dan *speedtest\_exporter* dapat diakses dan dianalisis secara terpusat melalui Prometheus.

c. *Import* Dasbor ke Grafana

1. Node Exporter Dashboard

Dasbor *node\_exporter* diimpor ke Grafana untuk memantau metrik sistem seperti penggunaan CPU, memori, *disk*, dan jaringan. Dasbor ini memberikan visualisasi yang komprehensif mengenai performa dan kesehatan sistem Raspberry Pi 4.

2. Ping Exporter Dashboard

Dasbor *ping\_exporter* diimpor ke Grafana untuk memantau latensi jaringan. Metrik yang diambil dari ping exporter mencakup latensi ping ke 8.8.8.8 dan google.com, memungkinkan pemantauan yang detail terhadap keterlambatan jaringan dan responsivitas.

3. Speedtest Exporter

Dasbor *speedtest\_exporter* diimpor ke Grafana untuk memantau kecepatan jaringan. Metrik dari speedtest exporter mencakup kecepatan unduh, kecepatan unggah, dan jitter, memberikan gambaran menyeluruh tentang kinerja jaringan.

4. Monitoring *Access Point* Dasbor

Dasbor utama ini dinamai "Monitoring Access Point" dan merupakan hasil dari modifikasi dan perluasan Node Exporter Dashboard. Pada dasbor ini, semua metrik dari *node\_exporter*, *ping\_exporter*, dan *speedtest\_exporter* digabungkan untuk memberikan visualisasi yang menyeluruh tentang performa dan kesehatan *access point* serta jaringan. Untuk mengakomodasi variabel-variabel berbeda dari dasbor *ping exporter* dan *speedtest exporter*, variabel-variabel ini ditambahkan ke dalam dasbor Monitoring Access Point. Modifikasi ini memungkinkan pemantauan terpusat dan komprehensif, serta memudahkan identifikasi dan analisis masalah spesifik pada masing-masing *node*. Dengan menggabungkan *query visualization* dari *node-node* lainnya ke dalam satu dasbor utama, pengguna dapat lebih mudah dan efisien memantau dan mengelola performa *access point* serta jaringan secara keseluruhan.

### 3.4 Kesimpulan Metode Penelitian

Bab 3 menjelaskan metodologi penelitian yang digunakan untuk mengembangkan sistem pemantauan jaringan di Universitas Islam Indonesia (UII) menggunakan teknologi *edge sensors*. Penelitian ini bertujuan untuk meningkatkan *robustness* dan *resilience* jaringan melalui implementasi teknis yang mendetail dan berorientasi pada efisiensi.

- a. Raspberry Pi 4 sebagai *Node Monitoring*: Raspberry Pi 4 dipilih sebagai *edge sensor* untuk mengumpulkan data metrik jaringan seperti *ping*, kecepatan unduh, kecepatan unggah, dan *traffic* jaringan. Alat ini dipasang pada titik strategis untuk memastikan cakupan pengumpulan data yang optimal. Raspberry Pi 4 dilengkapi dengan beberapa *exporter*, seperti:
  1. *Node Exporter* untuk mengukur penggunaan sumber daya sistem (CPU, RAM, dan jaringan).
  2. *Ping Exporter* untuk mencatat waktu respons dari berbagai titik *ping*, seperti ke server *DNS* publik (misalnya, Google *DNS* 8.8.8.8).
  3. *Speedtest Exporter* untuk mengukur kecepatan unduh dan unggah jaringan.
- b. Prometheus untuk Pengumpulan Data: Data yang dikumpulkan oleh Raspberry Pi 4 secara berkala dikirimkan ke Prometheus yang diinstal pada lingkungan Docker di sistem Windows. Prometheus digunakan untuk mengumpulkan, menyimpan, dan mengelola metrik dari berbagai sumber data secara *real-time*. Fail konfigurasi yang diatur pada Prometheus menentukan target pengumpulan data dari *node* di jaringan kampus.
- c. Grafana untuk Visualisasi Data: Data metrik yang dikumpulkan oleh Prometheus kemudian divisualisasikan menggunakan Grafana. Dasbor di Grafana diatur untuk menampilkan berbagai metrik performa jaringan dalam bentuk grafik yang interaktif dan mudah dipahami. Dasbor ini menampilkan:
  1. Waktu respons *ping* untuk mengukur latensi jaringan.
  2. Kecepatan unduh dan unggah untuk memastikan kualitas koneksi.
  3. Grafik *traffic* jaringan yang memberikan gambaran penggunaan *bandwidth* di berbagai titik jaringan.
- d. Integrasi Tailscale untuk Keamanan: Untuk memastikan konektivitas yang aman antara Raspberry Pi 4 dan server Prometheus di Docker, digunakan Tailscale sebagai alat VPN. Tailscale memungkinkan komunikasi yang aman antar perangkat di jaringan yang berbeda, memanfaatkan WireGuard untuk membangun koneksi *peer-to-peer* yang terenkripsi.

## BAB IV

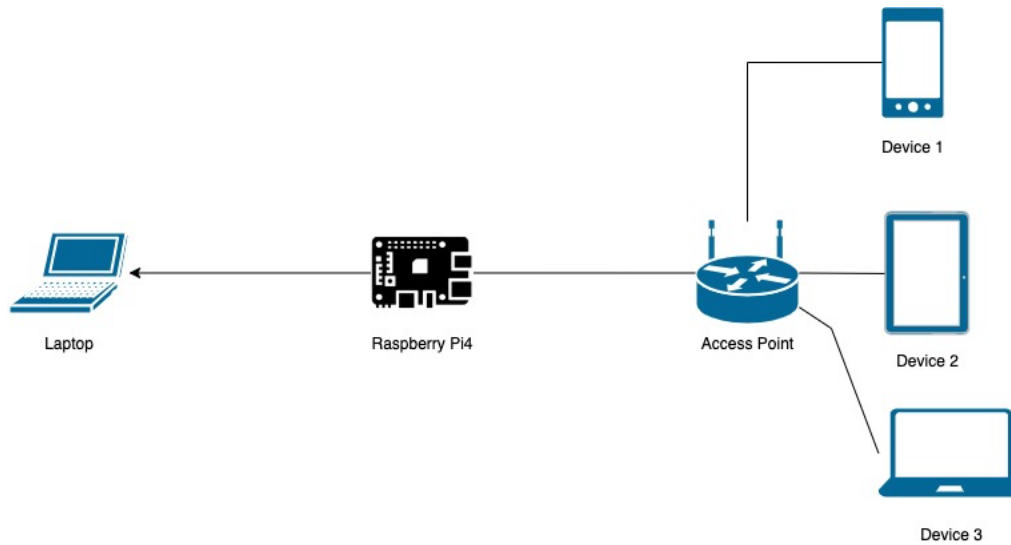
### IMPLEMENTASI DAN HASIL

Tahap ini menjelaskan tujuan dan pentingnya implementasi dalam penelitian. Fokusnya adalah untuk menunjukkan bagaimana implementasi ini membantu mencapai tujuan penelitian dan menyelesaikan masalah yang telah diidentifikasi dalam analisis kebutuhan.

#### 4.1 Topologi Pengambilan Data Access Point

Pada topologi ini, pengambilan data dari access point dilakukan melalui sebuah Raspberry Pi 4. Raspberry Pi 4 bertindak sebagai pengumpul data utama yang menangkap informasi-informasi terkait performa jaringan, seperti ping, packet loss, kecepatan unduh dan unggah, serta lalu lintas jaringan lainnya. Proses pengumpulan data ini memanfaatkan *node\_exporter* untuk pengumpulan data sistem, *speedtest\_exporter* untuk kecepatan jaringan, dan *ping\_exporter* untuk latensi jaringan.

Data yang terkumpul di Raspberry Pi 4 selanjutnya akan diteruskan ke laptop yang telah diinstal Prometheus dan Grafana. Prometheus berfungsi sebagai *database* yang mengelola dan menyimpan data-data dari Raspberry Pi 4. Setelah data tersimpan di Prometheus, Grafana akan mengambil data tersebut untuk divisualisasikan. Dengan visualisasi di Grafana, informasi terkait performa jaringan dari *access point* menjadi lebih mudah dipahami dan dipantau secara *real-time*. Topologi ini memungkinkan pemantauan jaringan yang terpusat dan *real-time* dengan memanfaatkan Raspberry Pi 4 sebagai *collector*, Prometheus sebagai penyimpan dan pengelola data, serta Grafana sebagai alat visualisasi.



Gambar 4.1 Topologi pengambila data *access point*

## 4.2 Penempatan Raspberry Pi 4

Raspberry Pi 4 yang telah dikonfigurasi untuk mengumpulkan data metrik jaringan ditempatkan di ruangan developer Badan Sistem Informasi, yang terletak di lantai 4 gedung rektorat Universitas Islam Indonesia dan ditempatkan juga di ruang kuliah umum lantai 1 sebelah barat utara Gedung Kuliah Umum Dr. Sardjito. Penempatan ini dipilih berdasarkan pertimbangan untuk mendapatkan jangkauan yang optimal terhadap *access point* yang terpasang di area tersebut. Dengan demikian, Raspberry Pi 4 dapat secara efektif mengumpulkan data yang diperlukan tanpa terhalang oleh masalah konektivitas atau jarak dari *access point*.

## 4.3 Pengumpulan Data

### 4.3.1 Pengumpulan Metrik

Raspberry Pi 4 telah diatur untuk mengumpulkan berbagai metrik jaringan penting, termasuk kecepatan unduh dan unggah (*speedtest*), waktu respon (*ping*), dan lalu lintas jaringan (*traffic*). Pengumpulan data ini dilakukan dengan menggunakan *node-node* yang telah diinstal sebelumnya, yaitu *node\_exporter*, *ping\_exporter*, dan *speedtest\_exporter*. Metrik ini dikirim ke Prometheus yang diinstal pada Docker di Windows, dan kemudian divisualisasikan di Grafana.

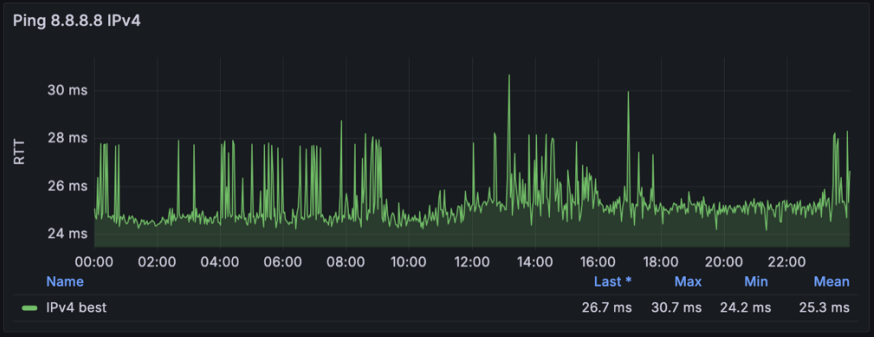
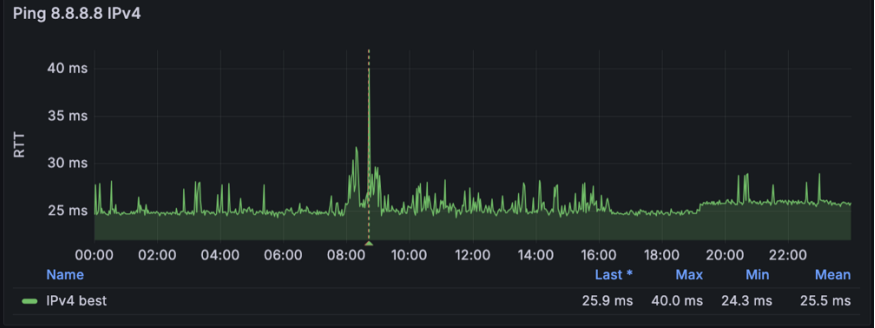
### 4.3.2 Frekuensi Pengambilan Data

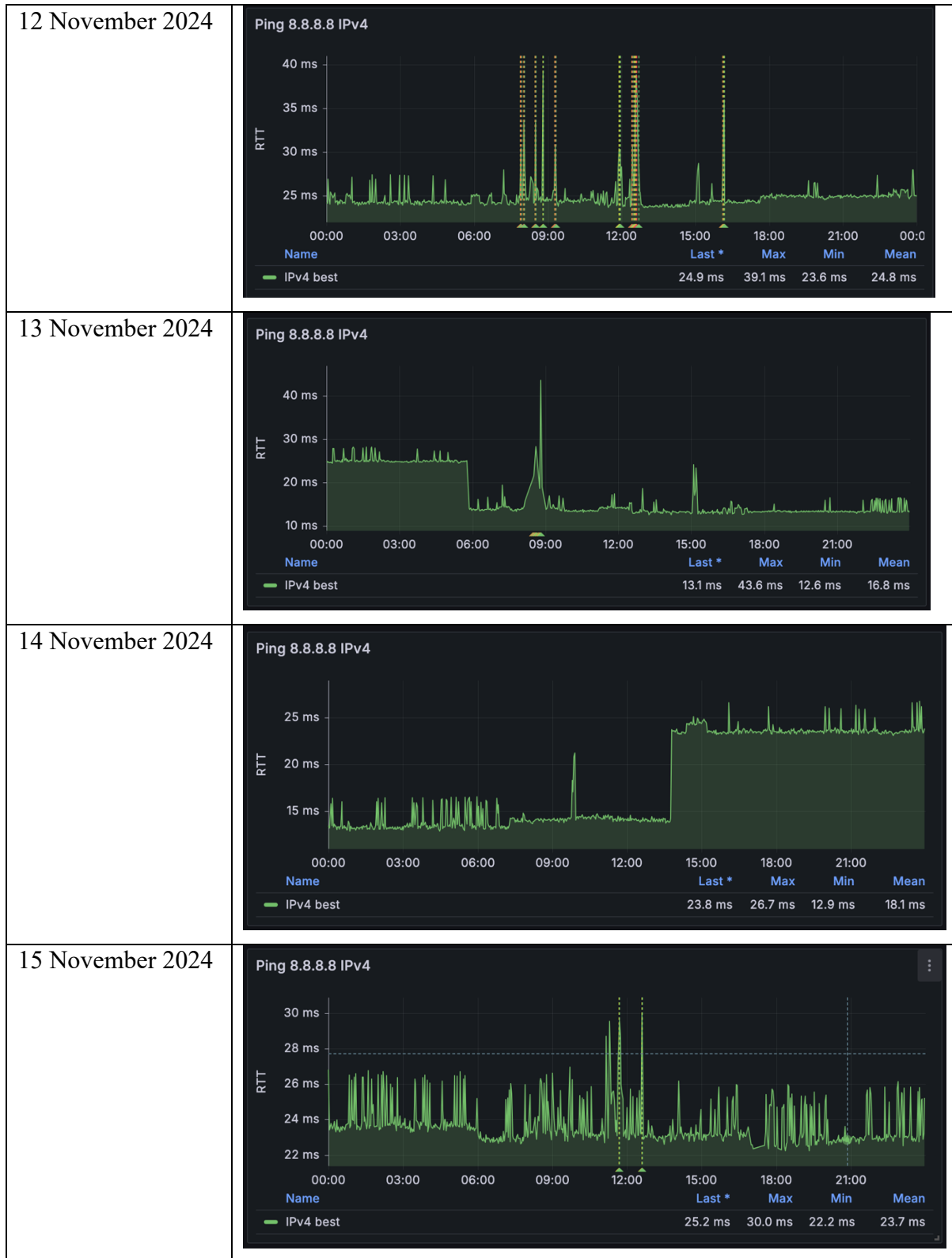
Untuk memastikan bahwa informasi yang diperoleh selalu terbaru dan relevan, pengumpulan data dilakukan secara berkala dengan interval waktu yang telah ditentukan. Hal ini memungkinkan pemantauan terus-menerus terhadap kinerja jaringan dan deteksi dini terhadap masalah yang mungkin timbul. Frekuensi pengambilan data diatur untuk memberikan keseimbangan antara kelengkapan data dan penggunaan sumber daya sistem yang efisien.

### 4.3.3 Implementasi Pengumpulan Data

Raspberry Pi 4 ditempatkan di ruangan developer Badan Sistem Informasi di lantai 4 gedung rektorat Universitas Islam Indonesia pada 2 September 2024. Data jaringan yang tercantum dalam Tabel 4.1. Meskipun hanya dilakukan pada dua tanggal tersebut, data yang terkumpul sudah cukup untuk menghasilkan visualisasi yang jelas pada dasbor Grafana. Data ini memberikan gambaran tentang kinerja jaringan di area tersebut, sehingga peneliti dapat melakukan analisis lebih lanjut untuk mengidentifikasi dan menangani masalah jaringan secara efektif.

Tabel 4.1 *Ping* pada tanggal 3, 5 September dan 12, 13, 14, 15 November 2024

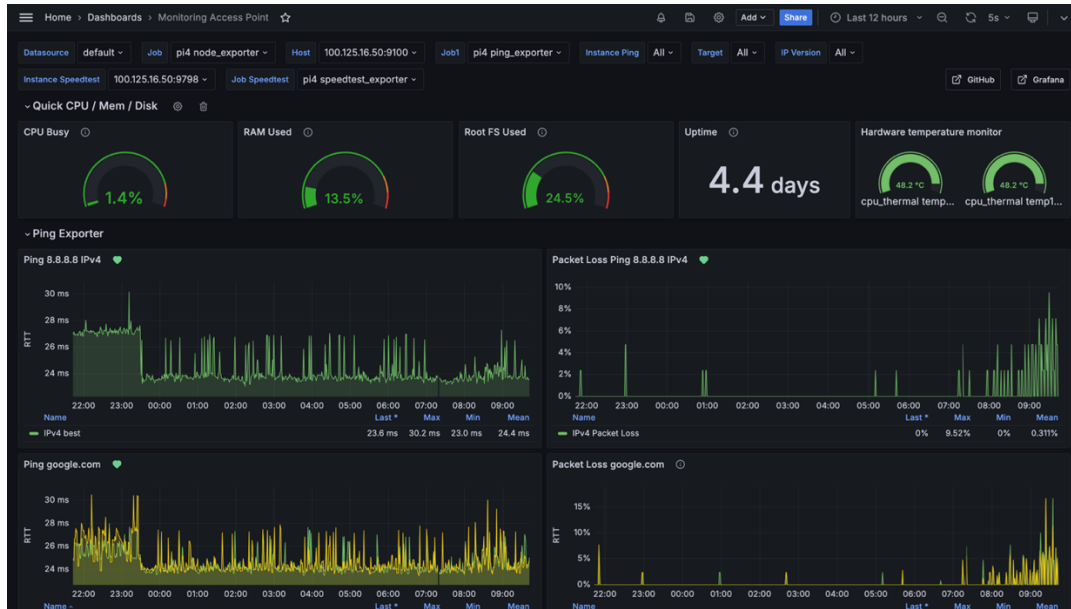
Tanggal	Grafik
3 September 2024	
5 September 2024	



#### 4.4 Pengujian pada Kondisi Nyata

Pada tahap ini, sistem pemantauan diuji pada jaringan yang sebenarnya untuk memastikan bahwa semua komponen bekerja sesuai rencana. Raspberry Pi 4 ditempatkan di

lingkungan Universitas Islam Indonesia dan dihubungkan ke *access point* untuk mengumpulkan data tentang kecepatan unduh, kecepatan unggah, *ping*, *packet loss*, dan *jitter*. Selama periode pengujian, sistem berjalan dengan lancar, mengumpulkan dan mengirim data ke Prometheus serta memvisualisasikannya di dasbor Grafana seperti pada Gambar 4.2



Gambar 4.2 Dasbor Grafana

## 4.5 Analisis Variabilitas Ping Menggunakan Standar Deviasi dan Rata-rata Deviasi

Dalam menganalisis performa jaringan, tidak hanya penting untuk mengetahui rata-rata waktu respons *ping*, tetapi juga variabilitas atau konsistensi dari nilai tersebut. Dua ukuran statistik yang umum digunakan untuk mengukur variabilitas ini adalah Rata-rata Deviasi (*Average Deviation*) dan Standar Deviasi (*Standard Deviation*). Pada bab ini, akan dilakukan perhitungan dan interpretasi terhadap kedua ukuran tersebut berdasarkan data waktu respons *ping* pada tujuh tanggal berbeda, yaitu 3 September 2024, 5 September 2024 pada ruang Developer BSI UII lalu dilakukan pengujian pada ruang kelas umum di Gedung Kuliah Umum Dr. Sadjito UII pada tanggal 12 November 2024 hingga 15 November 2024.

### 4.5.1 Data dan Perhitungan

Berikut Tabel 4.2 hingga Tabel 4.7 adalah data waktu respons *ping* pada tanggal 3, 4 September 2024 dan 12, 13, 14, 15 November 2024 sebagai berikut:

Tabel 4.2 Ping pada 3 September 2024


3 September 2024	
Waktu dan rata-rata	Screenshot Grafana
00:00 - 08:00 = 25.1 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>29 ms 28 ms 27 ms 26 ms 25 ms 24 ms</p> <p>00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00 08:00</p> <p>Name Last * Max Min Mean</p> <p>IPv4 best 25.7 ms 28.8 ms 24.2 ms 25.1 ms</p>
08:00 - 09:00 = 25.4 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>29 ms 28 ms 27 ms 26 ms 25 ms 24 ms</p> <p>08:00 08:05 08:10 08:15 08:20 08:25 08:30 08:35 08:40 08:45 08:50 08:55 09:00</p> <p>Name Last * Max Min Mean</p> <p>IPv4 best 25.1 ms 28.5 ms 24.3 ms 25.4 ms</p>
09:00 - 11:30 = 24.9 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>29 ms 28 ms 27 ms 26 ms 25 ms 24 ms</p> <p>09:00 09:15 09:30 09:45 10:00 10:15 10:30 10:45 11:00 11:15 11:30</p> <p>Name Last * Max Min Mean</p> <p>IPv4 best 24.8 ms 28.6 ms 24.2 ms 24.9 ms</p>
11:30 - 13:30 = 25.6 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>30 ms 28 ms 26 ms 24 ms</p> <p>11:30 11:40 11:50 12:00 12:10 12:20 12:30 12:40 12:50 13:00 13:10 13:20 13:30</p> <p>Name Last * Max Min Mean</p> <p>IPv4 best 26.6 ms 31.2 ms 24.4 ms 25.6 ms</p>
13:30 - 14:30 = 25.8 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>30 ms 28 ms 26 ms 24 ms</p> <p>13:30 13:35 13:40 13:45 13:50 13:55 14:00 14:05 14:10 14:15 14:20 14:25 14:30</p> <p>Name Last * Max Min Mean</p> <p>IPv4 best 27.7 ms 29.7 ms 24.4 ms 25.8 ms</p>

14:30 - 16:00 = 25.8 ms	
16:00 - 23:59:59 = 25.3 ms	

Pada tanggal 3 September 2024, performa jaringan menunjukkan stabilitas yang sangat baik dengan rata-rata waktu respons berkisar antara 24.9 ms hingga 25.8 ms sepanjang hari. Performa optimal terjadi pada rentang waktu 09:00 - 11:30, di mana rata-rata waktu respons mencapai 24.9 ms, mencerminkan kondisi jaringan yang sangat optimal. Meskipun terdapat sedikit peningkatan waktu respons pada periode 13:30 - 16:00 dengan rata-rata 25.8 ms, perubahan ini masih berada dalam batas normal dan tidak mengindikasikan gangguan signifikan.

Secara keseluruhan, grafik tidak menunjukkan adanya lonjakan besar yang dapat memengaruhi pengalaman pengguna. Stabilitas tinggi yang terlihat pada hari ini menunjukkan bahwa konfigurasi jaringan yang digunakan sudah optimal.

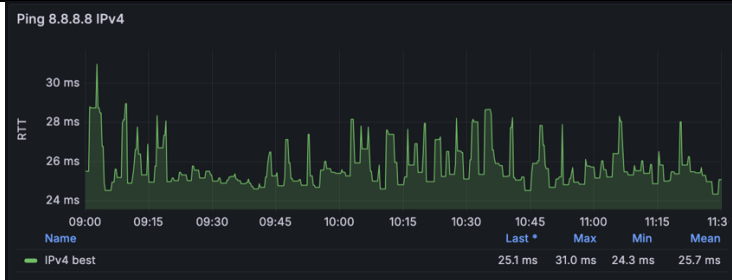
Tabel 4.3 Ping pada 5 September 2024

5 September 2024	
Waktu dan rata-rata	Screenshot Grafana
00:00 - 08:00 = 25 ms	

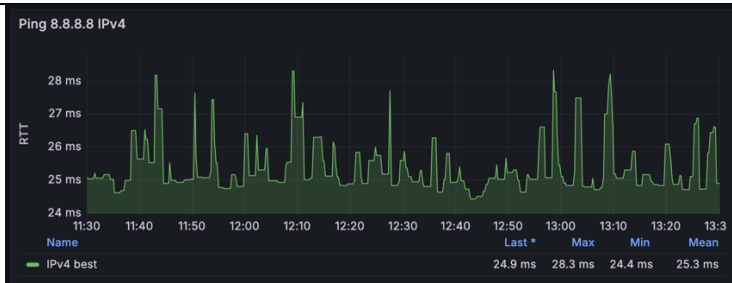
08:00 - 09:00 = 27.6 ms



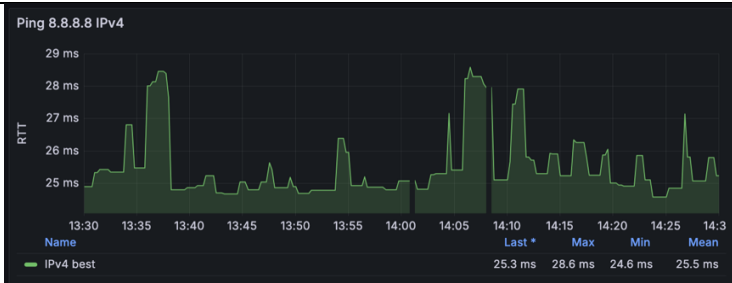
09:00 - 11:30 = 25.7 ms



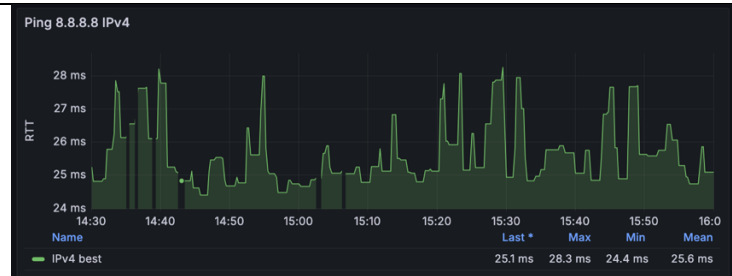
11:30 - 13:30 = 25.3 ms



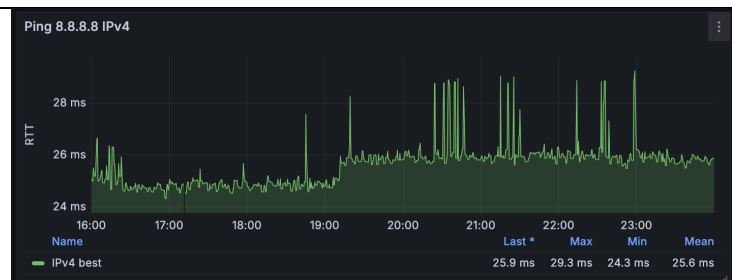
13:30 - 14:30 = 25.5 ms



14:30 - 16:00 = 25.6 ms




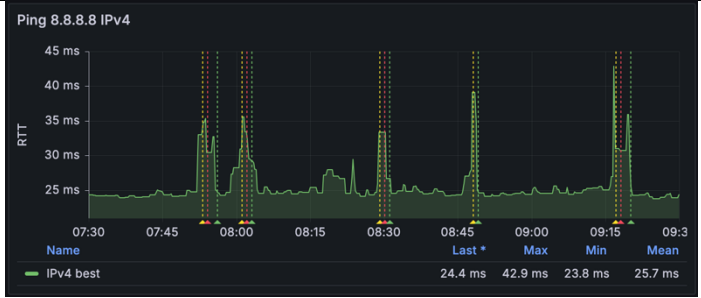
16:00 - 23:59:59 = 25.6 ms

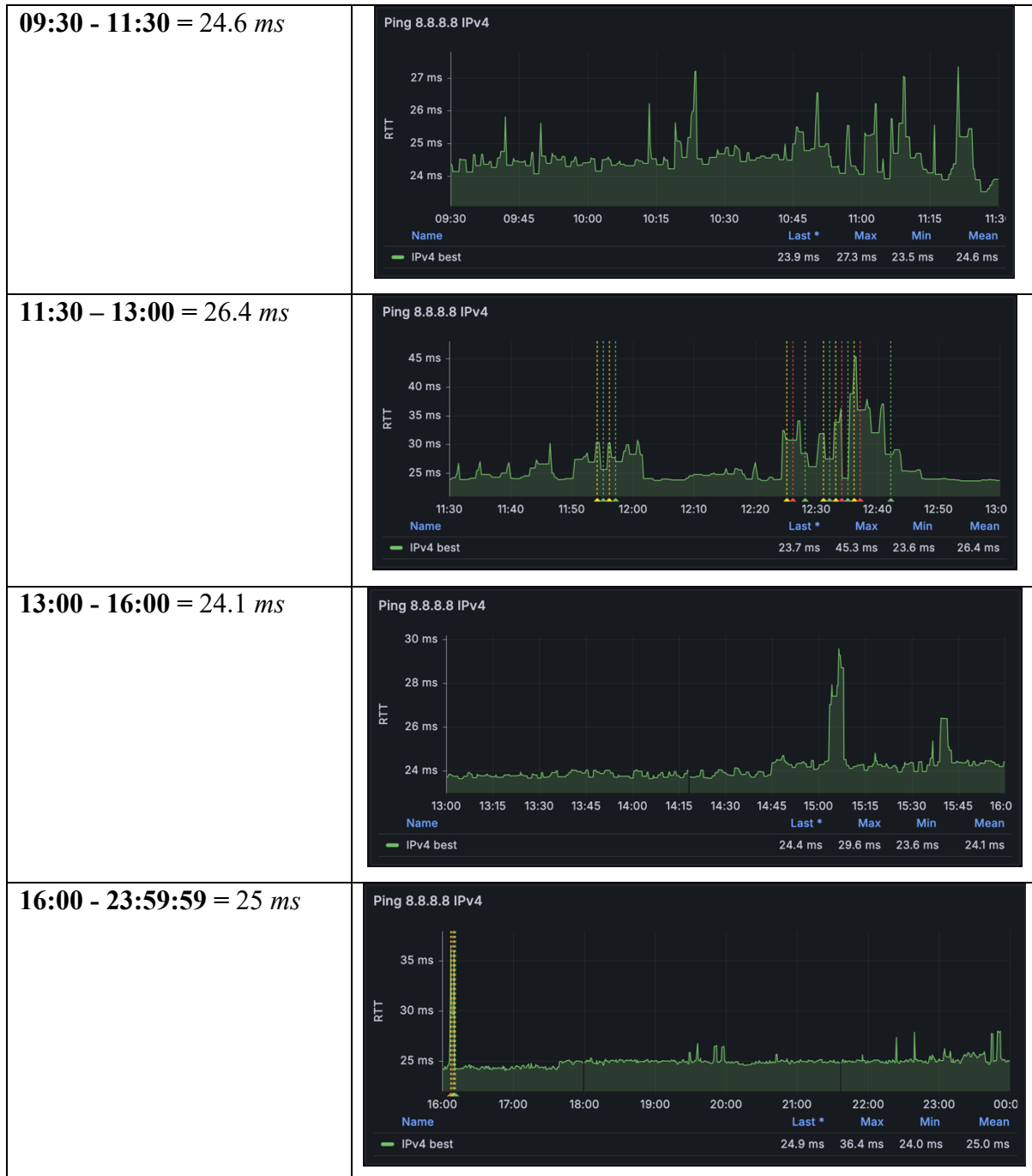


Pada tanggal 5 September 2024, performa jaringan menunjukkan stabilitas yang cukup baik dengan rata-rata waktu respons berkisar antara 25 ms hingga 27.6 ms sepanjang hari. Rentang waktu dengan waktu respons terbaik adalah pada periode 00:00 - 08:00, dengan rata-rata 25 ms, menunjukkan kondisi jaringan yang sangat stabil. Sementara itu, waktu respons tertinggi terjadi pada rentang 08:00 - 09:00, dengan rata-rata mencapai 27.6 ms, yang dapat mengindikasikan peningkatan beban jaringan atau aktivitas yang lebih intens.

Secara keseluruhan, grafik tidak menunjukkan lonjakan waktu respons yang signifikan di luar rentang normal, dengan stabilitas yang kembali pulih pada periode setelah pukul 09:00 hingga akhir hari. Variasi waktu respons tetap berada dalam batas yang wajar, dengan sedikit fluktuasi antara 25.3 ms hingga 25.6 ms pada siang hingga malam hari. Kondisi jaringan pada tanggal ini tergolong baik dan tidak membutuhkan intervensi teknis, meskipun peningkatan performa pada pagi hari dapat menjadi fokus untuk optimasi lebih lanjut.

Tabel 4.4 *Ping* pada 12 November 2024

12 November 2024	
Waktu dan rata-rata	Screenshot Grafana
00:00 - 07:30 = 24.5 ms	
07:30 - 09:30 = 25.7 ms	



Data *ping* yang diambil dari ruang kuliah Gedung Dr. Sardjito UII menunjukkan performa jaringan yang cukup stabil dengan RTT rata-rata berkisar antara 24,1 *ms* hingga 26,4 *ms*. Secara umum, jaringan mampu menjaga latensi rendah, namun terdapat beberapa lonjakan yang signifikan, khususnya pada waktu-waktu tertentu seperti pukul 08:00, 12:30, dan 15:00, dengan RTT mencapai hingga 45 *ms*. Lonjakan ini kemungkinan terjadi akibat tingginya aktivitas pengguna, seperti perkuliahan atau akses simultan pada jaringan.

Tabel 4.5 Ping pada 13 November 2024

13 November 2024											
Waktu dan rata-rata	Screenshot Grafana										
00:00 - 07:30 = 22.8 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Last *</th> <th>Max</th> <th>Min</th> <th>Mean</th> </tr> </thead> <tbody> <tr> <td>IPv4 best</td> <td>14.4 ms</td> <td>36.2 ms</td> <td>13.5 ms</td> <td>22.8 ms</td> </tr> </tbody> </table>	Name	Last *	Max	Min	Mean	IPv4 best	14.4 ms	36.2 ms	13.5 ms	22.8 ms
Name	Last *	Max	Min	Mean							
IPv4 best	14.4 ms	36.2 ms	13.5 ms	22.8 ms							
07:30 - 09:30 = 24.9 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>07:30 07:45 08:00 08:15 08:30 08:45 09:00 09:15 09:30</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Last *</th> <th>Max</th> <th>Min</th> <th>Mean</th> </tr> </thead> <tbody> <tr> <td>IPv4 best</td> <td>13.8 ms</td> <td>1.08 s</td> <td>13.4 ms</td> <td>24.9 ms</td> </tr> </tbody> </table>	Name	Last *	Max	Min	Mean	IPv4 best	13.8 ms	1.08 s	13.4 ms	24.9 ms
Name	Last *	Max	Min	Mean							
IPv4 best	13.8 ms	1.08 s	13.4 ms	24.9 ms							
09:30 - 11:30 = 13.8 ms	<p>Ping 8.8.8.8 IPv4</p> <p>Packet Loss Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>09:30 09:45 10:00 10:15 10:30 10:45 11:00 11:15 11:30</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Last *</th> <th>Max</th> <th>Min</th> <th>Mean</th> </tr> </thead> <tbody> <tr> <td>IPv4 best</td> <td>14.1 ms</td> <td>16.9 ms</td> <td>13.2 ms</td> <td>13.8 ms</td> </tr> </tbody> </table>	Name	Last *	Max	Min	Mean	IPv4 best	14.1 ms	16.9 ms	13.2 ms	13.8 ms
Name	Last *	Max	Min	Mean							
IPv4 best	14.1 ms	16.9 ms	13.2 ms	13.8 ms							
11:30 - 13:00 = 14.2 ms	<p>Ping 8.8.8.8 IPv4</p> <p>RTT</p> <p>11:30 11:40 11:50 12:00 12:10 12:20 12:30 12:40 12:50 13:00</p> <table border="1"> <thead> <tr> <th>Name</th> <th>Last *</th> <th>Max</th> <th>Min</th> <th>Mean</th> </tr> </thead> <tbody> <tr> <td>IPv4 best</td> <td>18.7 ms</td> <td>18.7 ms</td> <td>12.9 ms</td> <td>14.2 ms</td> </tr> </tbody> </table>	Name	Last *	Max	Min	Mean	IPv4 best	18.7 ms	18.7 ms	12.9 ms	14.2 ms
Name	Last *	Max	Min	Mean							
IPv4 best	18.7 ms	18.7 ms	12.9 ms	14.2 ms							



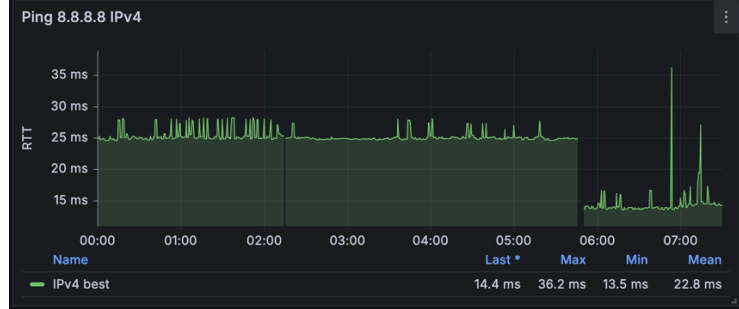
Berdasarkan data yang diambil pada tanggal Rabu, 13 November 2024, di ruang kuliah Gedung Dr. Sardjito UII, grafik menunjukkan performa jaringan terhadap server *DNS* publik Google (8.8.8.8). Secara umum, rata-rata latensi (RTT) berada pada kisaran 13 ms hingga 25 ms, yang mencerminkan koneksi jaringan yang stabil dan responsif untuk sebagian besar waktu.

Namun, terdapat beberapa anomali signifikan. Pada grafik pagi hari sekitar pukul 07:45 hingga 08:30, terjadi lonjakan latensi yang cukup besar, bahkan mencapai 1 detik. Lonjakan ini kemungkinan besar disebabkan oleh tingginya beban jaringan atau aktivitas simultan di jaringan kampus, seperti pengguna yang mengakses internet secara bersamaan. Setelah waktu tersebut, latensi kembali stabil pada kisaran rata-rata sekitar 13 ms hingga 16 ms.

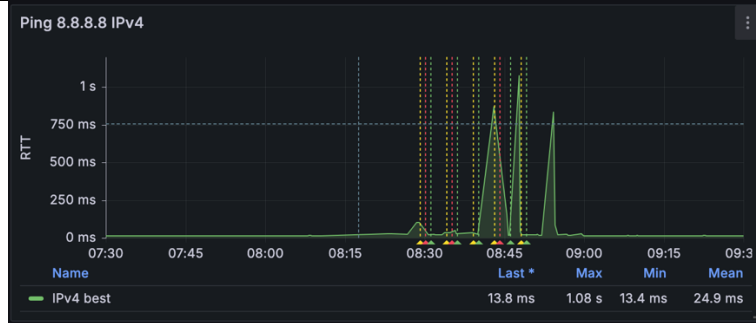
Tabel 4.6 *Ping* pada 14 November 2024

14 November 2024	
Waktu dan rata-rata	<i>Screenshot Grafana</i>

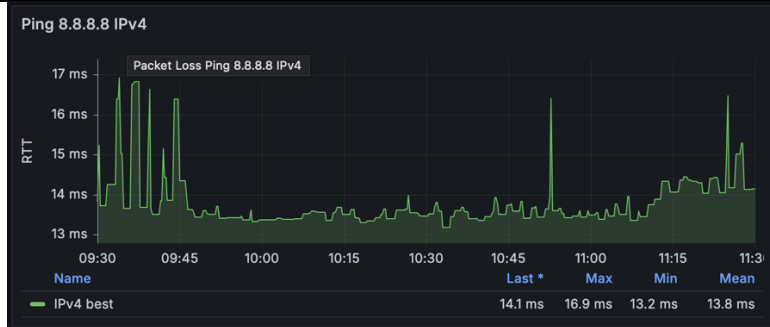
00:00 - 07:30 = 13.8 ms



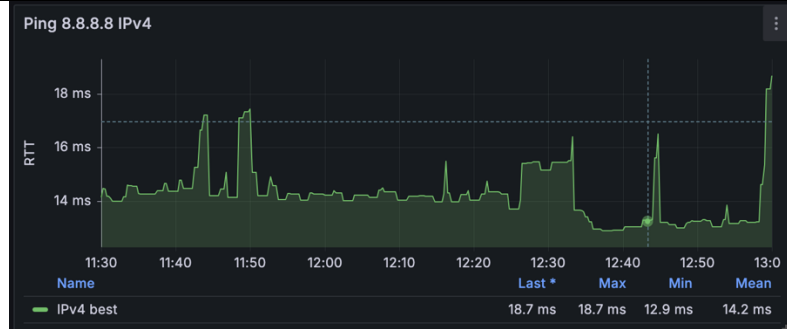
07:30 - 09:30 = 14 ms



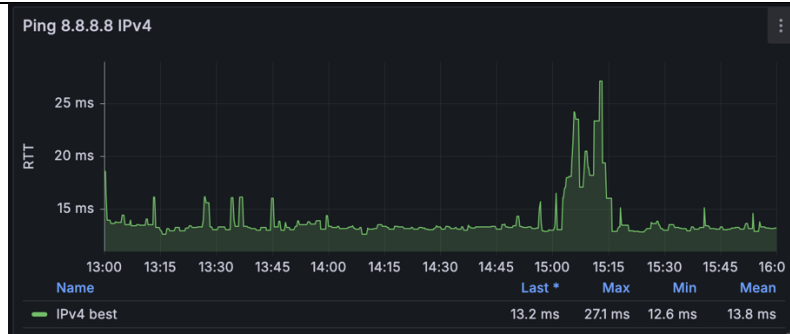
09:30 - 11:30 = 14.6 ms

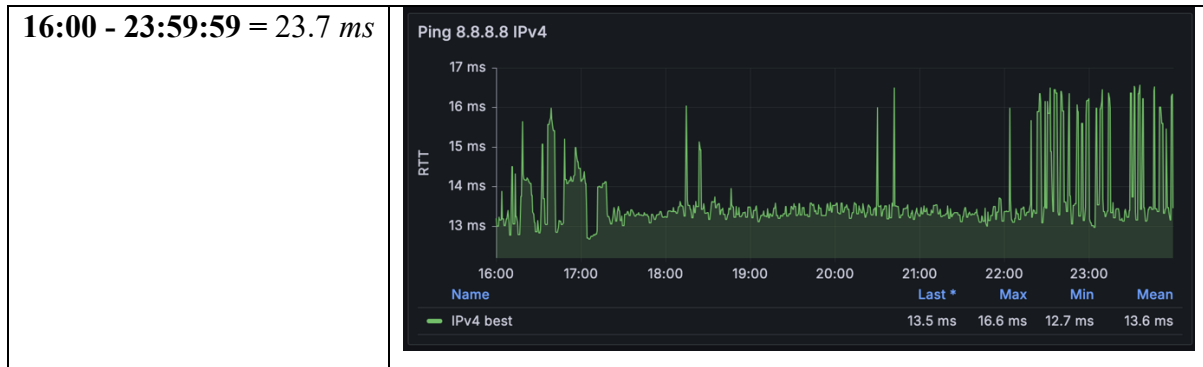


11:30 - 13:00 = 14.1 ms



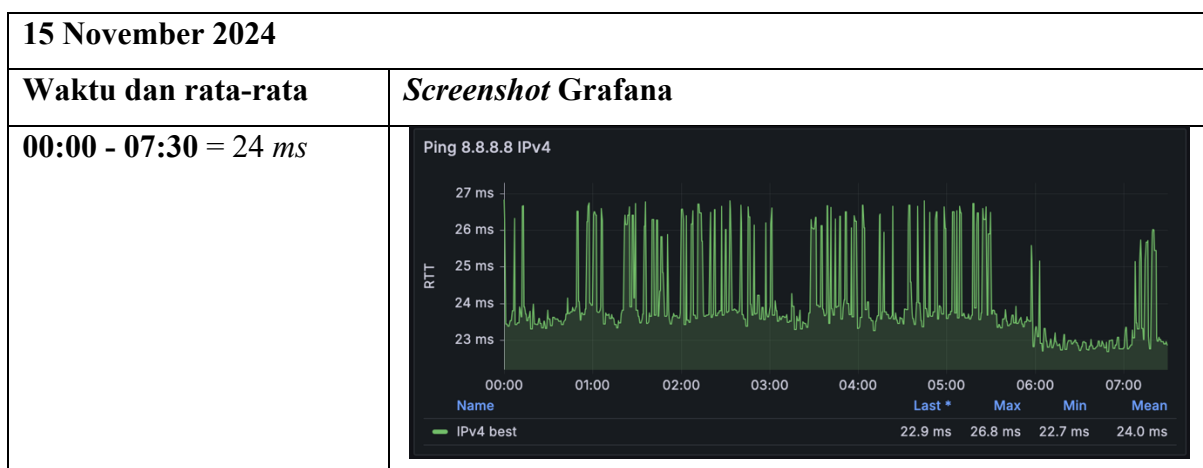
13:00 - 16:00 = 21.3 ms



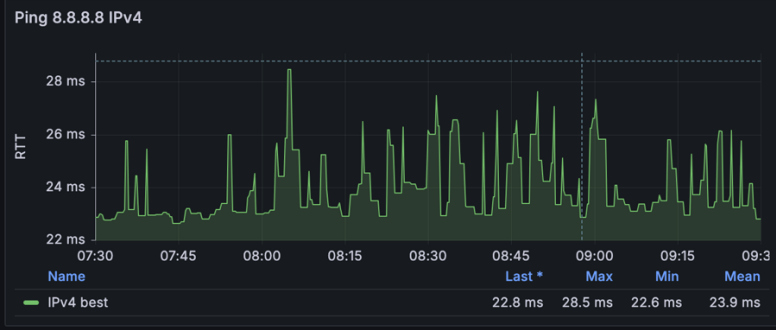


Pada tanggal Kamis, 14 November 2024, dilakukan pengukuran performa jaringan di ruang kuliah Gedung Dr. Sardjito UII terhadap server *DNS* publik Google (8.8.8.8). Grafik menunjukkan rata-rata RTT berada dalam kisaran 13 ms hingga 23 ms, yang mencerminkan stabilitas jaringan yang baik dan responsif selama sebagian besar waktu pengamatan. Lonjakan latensi cenderung jarang terjadi, dengan beberapa anomali kecil pada waktu tertentu. Salah satu lonjakan terlihat pada siang hari, sekitar pukul 13:30 hingga 15:30, di mana rata-rata RTT meningkat ke kisaran 23 ms hingga 27 ms, yang kemungkinan diakibatkan oleh peningkatan aktivitas jaringan atau kemacetan sementara. Selain itu, pada malam hari sekitar pukul 20:00 hingga 22:00, terdapat lonjakan-lonjakan kecil dengan RTT mencapai 26 ms, namun jaringan dengan cepat pulih kembali ke rata-rata yang stabil.

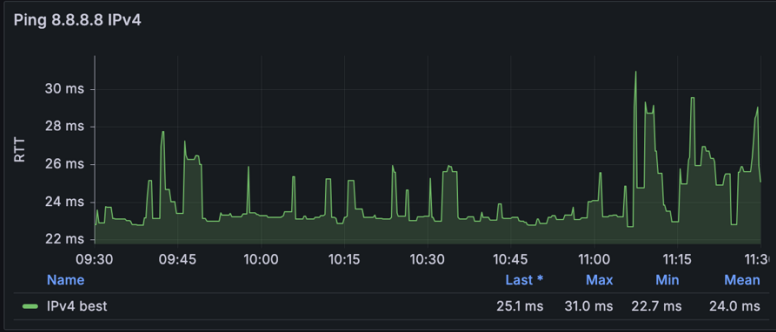
Tabel 4.7 *Ping* pada 15 November 2024



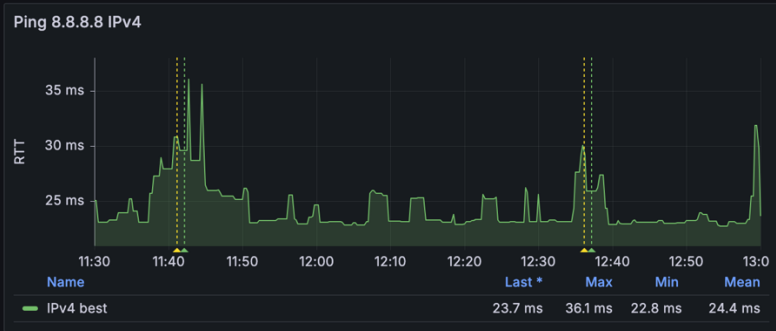
**07:30 - 09:30 = 23.9 ms**



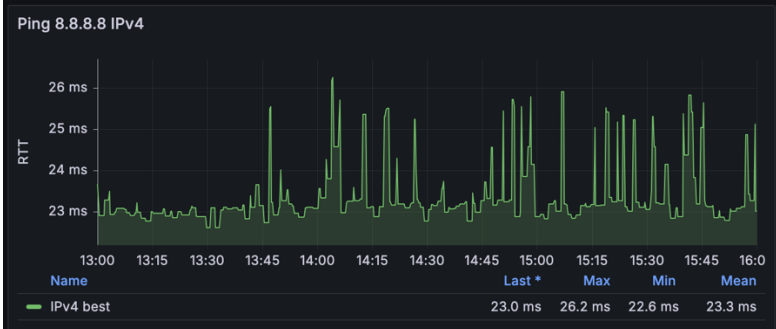
**09:30 - 11:30 = 24 ms**



**11:30 - 13:00 = 24.4 ms**



**13:00 - 16:00 = 23.3 ms**



**16:00 - 23:59:59 = 23.5 ms**



Pada tanggal Jumat, 15 November 2024, data yang diambil di ruang kuliah Gedung Dr Sardjito UII menunjukkan performa jaringan yang cukup stabil secara keseluruhan, dengan rata-rata latensi ke server 8.8.8.8 (Google DNS) berada di kisaran 23-24 ms. Pada periode dini hari hingga pagi (00:00 - 07:00), jaringan menunjukkan stabilitas tinggi dengan beberapa fluktuasi kecil, yang mencerminkan aktivitas jaringan yang rendah. Memasuki pagi hingga siang hari (07:30 - 12:30), terlihat peningkatan fluktuasi latensi, terutama dengan puncak tertinggi mencapai 36 ms pada pukul 11:40. Lonjakan ini kemungkinan diakibatkan oleh tingginya aktivitas jaringan dari pengguna, seperti mahasiswa dan staf yang menggunakan internet untuk kegiatan belajar dan administrasi.

Pada siang hingga sore hari (13:00 - 16:00), latensi cenderung stabil dengan fluktuasi kecil yang tidak signifikan, meskipun terdapat lonjakan kecil sesaat sekitar pukul 15:30. Hal ini menunjukkan bahwa aktivitas jaringan pada periode ini lebih merata, kemungkinan terkait dengan kegiatan rutin yang tidak terlalu membebani jaringan. Pada sore hingga malam hari (16:00 - 23:59), grafik menunjukkan fluktuasi yang lebih teratur dengan beberapa lonjakan hingga 27 ms. Periode ini sering kali mencerminkan aktivitas yang lebih bervariasi, seperti streaming, komunikasi, atau tugas-tugas berbasis internet.

#### 4.5.2 Perhitungan untuk 3 September 2024

##### a. Rata-rata (*Mean*)

Rata-rata memberikan gambaran umum mengenai waktu respons *ping* selama jangka waktu tertentu. Rumus yang digunakan untuk menghitung rata-rata adalah sebagai berikut:

$$\text{Mean} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.1)$$

$$\text{Mean} = \frac{25.1 + 25.4 + 24.9 + 25.6 + 25.8 + 25.8 + 25.3}{7} = 25.41 \text{ ms}$$

##### b. Rata-rata Deviasi

Deviasi rata-rata menggambarkan sejauh mana setiap pengukuran berbeda dari rata-rata. Ini dihitung dengan mengambil rata-rata dari perbedaan absolut antara setiap nilai dan rata-rata. Rumus deviasi rata-rata adalah:

$$\text{Average Deviation} = \frac{\sum_{i=1}^n |x_i - \text{Mean}|}{n} \quad (4.2)$$

Avg Deviation

$$= \frac{|25.1 - 25.4| + |25.4 - 25.4| + |24.9 - 25.4| + |25.6 - 25.4| + |25.8 - 25.4| + |25.8 - 25.4| + |25.3 - 25.4|}{7}$$

$$\text{Avg Deviation} = 0.273 \text{ ms}$$

c. Standar Deviasi

Deviasi standar mengukur sejauh mana data tersebar dari rata-rata, memberikan wawasan lebih mendalam mengenai variasi dalam kinerja jaringan. Rumus untuk deviasi standar sebagai berikut:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n}} \quad (4.3)$$

$$\sqrt{\frac{(25.1 - 25.4)^2 + (25.4 - 25.4)^2 + (24.9 - 25.4)^2 + (25.6 - 25.4)^2 + (25.8 - 25.4)^2 + (25.8 - 25.4)^2 + (25.3 - 25.4)^2}{7}}$$

$$\text{Standard Deviation} = 0.318 \text{ ms}$$

### 4.5.3 Perhitungan untuk 5 September 2024

a. Rata-rata (*Mean*)

Rata-rata memberikan gambaran umum mengenai waktu respons *ping* selama jangka waktu tertentu. Rumus yang digunakan untuk menghitung rata-rata adalah sebagai berikut:

$$\text{Mean} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.4)$$

$$\text{Mean} = \frac{25.0 + 27.6 + 25.7 + 25.3 + 25.5 + 25.6 + 25.6}{7} = 25.76 \text{ ms}$$

b. Rata-rata Deviasi

Deviasi rata-rata menggambarkan sejauh mana setiap pengukuran berbeda dari rata-rata. Ini dihitung dengan mengambil rata-rata dari perbedaan absolut antara setiap nilai dan rata-rata. Rumus deviasi rata-rata adalah:

$$\text{Average Deviation} = \frac{\sum_{i=1}^n |x_i - \text{Mean}|}{n} \quad (4.5)$$

Avg Deviation

$$= \frac{|25.0 - 25.76| + |27.6 - 25.76| + |25.7 - 25.3| + |25.3 - 25.76| + |25.5 - 25.76| + |25.6 - 25.76| + |25.6 - 25.76|}{7}$$

$$\text{Avg Deviation} = 0.527 \text{ ms}$$

#### c. Standar Deviasi

Deviasi standar mengukur sejauh mana data tersebar dari rata-rata, memberikan wawasan lebih mendalam mengenai variasi dalam kinerja jaringan. Rumus untuk deviasi standar sebagai berikut:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n}} \quad (4.6)$$

$$\sqrt{\frac{(25.0 - 25.76)^2 + (27.6 - 25.76)^2 + (25.7 - 25.3)^2 + (25.3 - 25.76)^2 + (25.5 - 25.76)^2 + (25.6 - 25.76)^2 + (25.6 - 25.76)^2}{7}}$$

$$\text{Standard Deviation} = 0.784 \text{ ms}$$

### 4.5.4 Perhitungan untuk 12 November 2024

#### a. Rata-rata (*Mean*)

Rata-rata memberikan gambaran umum mengenai waktu respons *ping* selama jangka waktu tertentu. Rumus yang digunakan untuk menghitung rata-rata adalah sebagai berikut:

$$\text{Mean} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.7)$$

$$\text{Mean} = \frac{24.5 + 25.7 + 24.6 + 26.4 + 24.1 + 25.0}{6} = 25.05 \text{ ms}$$

b. Rata-rata Deviasi

Deviasi rata-rata menggambarkan sejauh mana setiap pengukuran berbeda dari rata-rata. Ini dihitung dengan mengambil rata-rata dari perbedaan absolut antara setiap nilai dan rata-rata. Rumus deviasi rata-rata adalah:

$$\text{Average Deviation} = \frac{\sum_{i=1}^n |x_i - \text{Mean}|}{n} \quad (4.8)$$

$$\begin{aligned} &\text{Avg Deviation} \\ &= \frac{|24.5 - 25.05| + |25.7 - 25.05| + |24.6 - 25.05| + |26.4 - 25.05| + |24.1 - 25.05| + |25.0 - 25.05|}{6} \end{aligned}$$

$$\text{Avg Deviation} = 0.67 \text{ ms}$$

c. Standar Deviasi

Deviasi standar mengukur sejauh mana data tersebar dari rata-rata, memberikan wawasan lebih mendalam mengenai variasi dalam kinerja jaringan. Rumus untuk deviasi standar sebagai berikut:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n}} \quad (4.9)$$

$$\sqrt{\frac{(24.5 - 25.05)^2 + (25.7 - 25.05)^2 + (24.6 - 25.05)^2 + (26.4 - 25.05)^2 + (24.1 - 25.05)^2 + (25.0 - 25.05)^2}{6}}$$

$$\text{Standard Deviation} = 0.78 \text{ ms}$$

#### 4.5.5 Perhitungan untuk 13 November 2024

##### a. Rata-rata (*Mean*)

Rata-rata memberikan gambaran umum mengenai waktu respons *ping* selama jangka waktu tertentu. Rumus yang digunakan untuk menghitung rata-rata adalah sebagai berikut:

$$\text{Mean} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.10)$$

$$\text{Mean} = \frac{22.8 + 24.9 + 13.8 + 14.2 + 13.8 + 13.6}{6} = 25.05 \text{ ms}$$

##### b. Rata-rata Deviasi

Deviasi rata-rata menggambarkan sejauh mana setiap pengukuran berbeda dari rata-rata. Ini dihitung dengan mengambil rata-rata dari perbedaan absolut antara setiap nilai dan rata-rata. Rumus deviasi rata-rata adalah:

$$\text{Average Deviation} = \frac{\sum_{i=1}^n |x_i - \text{Mean}|}{n} \quad (4.11)$$

$$\text{Avg Deviation} = \frac{|22.8 - 17.18| + |24.9 - 17.18| + |13.8 - 17.18| + |14.2 - 17.18| + |13.8 - 17.18| + |13.6 - 17.18|}{6}$$

$$\text{Avg Deviation} = 4.44 \text{ ms}$$

##### c. Standar Deviasi

Deviasi standar mengukur sejauh mana data tersebar dari rata-rata, memberikan wawasan lebih mendalam mengenai variasi dalam kinerja jaringan. Rumus untuk deviasi standar sebagai berikut:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n}} \quad (4.12)$$

$$\sqrt{\frac{(22.8 - 17.18)^2 + (24.9 - 17.18)^2 + (13.8 - 17.18)^2 + (14.2 - 17.18)^2 + (13.8 - 17.18)^2 + (13.6 - 17.18)^2}{6}}$$

$$\text{Standard Deviation} = 4.76 \text{ ms}$$

#### 4.5.6 Perhitungan untuk 14 November 2024

##### a. Rata-rata (*Mean*)

Rata-rata memberikan gambaran umum mengenai waktu respons *ping* selama jangka waktu tertentu. Rumus yang digunakan untuk menghitung rata-rata adalah sebagai berikut:

$$\text{Mean} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.73)$$

$$\text{Mean} = \frac{13.8 + 14.0 + 14.6 + 14.1 + 21.3 + 23.7}{6} = 16.92 \text{ ms}$$

##### b. Rata-rata Deviasi

Deviasi rata-rata menggambarkan sejauh mana setiap pengukuran berbeda dari rata-rata. Ini dihitung dengan mengambil rata-rata dari perbedaan absolut antara setiap nilai dan rata-rata. Rumus deviasi rata-rata adalah:

$$\text{Average Deviation} = \frac{\sum_{i=1}^n |x_i - \text{Mean}|}{n} \quad (4.14)$$

$$\text{Avg Deviation} = \frac{|13.8 - 16.92| + |14.0 - 16.92| + |14.6 - 16.92| + |14.1 - 16.92| + |21.3 - 16.92| + |23.7 - 16.92|}{6}$$

$$\text{Avg Deviation} = 3.72 \text{ ms}$$

##### c. Standar Deviasi

Deviasi standar mengukur sejauh mana data tersebar dari rata-rata, memberikan wawasan lebih mendalam mengenai variasi dalam kinerja jaringan. Rumus untuk deviasi standar sebagai berikut:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n}} \quad (4.15)$$

$$\sqrt{\frac{(13.8 - 16.92)^2 + (14.0 - 16.92)^2 + (14.6 - 16.92)^2 + (14.1 - 16.92)^2 + (21.3 - 16.92)^2 + (23.7 - 16.92)^2}{6}}$$

$$\text{Standard Deviation} = 4.01 \text{ ms}$$

#### 4.5.7 Perhitungan untuk 15 November 2024

##### d. Rata-rata (*Mean*)

Rata-rata memberikan gambaran umum mengenai waktu respons *ping* selama jangka waktu tertentu. Rumus yang digunakan untuk menghitung rata-rata adalah sebagai berikut:

$$\text{Mean} = \frac{\sum_{i=1}^n x_i}{n} \quad (4.86)$$

$$\text{Mean} = \frac{24.0 + 23.9 + 24.0 + 24.4 + 23.3 + 23.5}{6} = 24.02 \text{ ms}$$

##### e. Rata-rata Deviasi

Deviasi rata-rata menggambarkan sejauh mana setiap pengukuran berbeda dari rata-rata. Ini dihitung dengan mengambil rata-rata dari perbedaan absolut antara setiap nilai dan rata-rata. Rumus deviasi rata-rata adalah:

$$\text{Average Deviation} = \frac{\sum_{i=1}^n |x_i - \text{Mean}|}{n} \quad (4.17)$$

$$\text{Avg Deviation} = \frac{|24.0 - 24.02| + |23.9 - 24.02| + |24.0 - 24.02| + |24.4 - 24.02| + |23.3 - 24.02| + |23.5 - 24.02|}{6}$$

$$\text{Avg Deviation} = 0.30 \text{ ms}$$

##### f. Standar Deviasi

Deviasi standar mengukur sejauh mana data tersebar dari rata-rata, memberikan wawasan lebih mendalam mengenai variasi dalam kinerja jaringan. Rumus untuk deviasi standar sebagai berikut:

$$\text{Standard Deviation} = \sqrt{\frac{\sum_{i=1}^n (x_i - \text{Mean})^2}{n}} \quad (4.18)$$

$$\sqrt{\frac{(24.0 - 24.02)^2 + (23.9 - 24.02)^2 + (24.0 - 24.02)^2 + (24.4 - 24.02)^2 + (23.3 - 24.02)^2 + (23.5 - 24.02)^2}{6}}$$

$$\text{Standard Deviation} = 0.40 \text{ ms}$$

#### 4.6 Interpretasi Hasil

Berdasarkan hasil analisis variabilitas *ping* menggunakan rata-rata deviasi dan standar deviasi, dapat disimpulkan bahwa performa jaringan pada kedua tanggal, yaitu 3 September 2024 dan 5 September 2024, menunjukkan karakteristik yang berbeda. Pada 3 September 2024, rata-rata deviasi sebesar 0.273 *ms* dan standar deviasi sebesar 0.318 *ms* mengindikasikan jaringan yang sangat stabil. Fluktuasi dalam waktu respons *ping* sangat kecil, sehingga jaringan dapat dianggap dalam kondisi baik dan konsisten, yang memberikan pengalaman penggunaan optimal, terutama untuk aplikasi yang sensitif terhadap latensi seperti komunikasi video atau video *streaming*.

Sementara itu, pada 5 September 2024, nilai rata-rata deviasi yang lebih tinggi, yaitu 0.527 *ms*, serta standar deviasi sebesar 0.784 *ms* menunjukkan adanya peningkatan variasi dalam waktu respons *ping*. Meskipun jaringan masih berfungsi dengan baik, variasi ini mengindikasikan potensi ketidakstabilan jaringan yang lebih besar dibandingkan dengan 3 September. Peningkatan variabilitas ini bisa disebabkan oleh peningkatan beban jaringan atau gangguan ringan, sehingga meskipun performa jaringan tetap dalam batas toleransi yang dapat diterima, stabilitas jaringan sedikit menurun pada hari tersebut.

Pada 12 November 2024, rata-rata waktu respons *ping* sebesar 25.05 *ms*, dengan rata-rata deviasi 0.67 *ms* dan standar deviasi 0.78 *ms*, menunjukkan stabilitas jaringan yang sangat baik. Fluktuasi dalam waktu respons *ping* yang kecil mencerminkan performa jaringan yang konsisten. Dengan kondisi ini, aktivitas yang membutuhkan latensi rendah seperti akses *Learning Management System (LMS)*, *streaming* video untuk perkuliahan daring, atau konferensi video dapat berjalan lancar tanpa gangguan. Kondisi ini mencerminkan kapasitas jaringan di ruangan tersebut mampu menangani beban penggunaan yang moderat dengan sangat baik.

Pada 13 November 2024, rata-rata waktu respons *ping* tetap 25.05 *ms*, namun dengan rata-rata deviasi yang lebih tinggi sebesar 4.44 *ms* dan standar deviasi 4.76 *ms*. Peningkatan variabilitas ini menunjukkan adanya fluktuasi yang signifikan dalam waktu respons jaringan, yang mungkin terjadi akibat penggunaan internet oleh lebih banyak perangkat atau aplikasi yang memakan *bandwidth* besar. Sebagai ruangan dengan kapasitas besar, lonjakan fluktuasi ini bisa saja disebabkan oleh banyaknya peserta yang terkoneksi secara bersamaan, misalnya pada saat sesi kuliah interaktif menggunakan perangkat daring atau *streaming* video. Meskipun demikian, jaringan tetap mampu berfungsi dengan baik untuk sebagian besar aktivitas. Namun,

potensi ketidakstabilan ini dapat memengaruhi aplikasi sensitif latensi seperti presentasi *live streaming* atau demonstrasi perangkat lunak.

Pada 14 November 2024, rata-rata waktu respons *ping* turun menjadi 16.92 ms, dengan rata-rata deviasi 3.72 ms dan standar deviasi 4.01 ms. Rata-rata waktu respons yang lebih rendah mengindikasikan performa jaringan yang cepat, namun adanya variabilitas yang cukup besar menunjukkan stabilitas yang sedikit menurun dibandingkan dengan 12 November. Ini dapat terjadi jika terdapat lonjakan aktivitas pada periode tertentu, misalnya saat banyak peserta mengunduh atau mengunggah fail besar secara bersamaan. Fluktuasi ini mungkin memberikan pengalaman yang kurang ideal untuk aplikasi yang memerlukan koneksi konsisten seperti kuliah daring berbasis *real-time*. Namun, untuk penggunaan umum seperti *browsing* atau akses dokumen, jaringan masih cukup mendukung.

Pada 15 November 2024, performa jaringan kembali sangat stabil, dengan rata-rata waktu respons 24.02 ms, rata-rata deviasi 0.30 ms, dan standar deviasi 0.40 ms. Nilai rata-rata deviasi dan standar deviasi yang rendah menunjukkan jaringan yang sangat konsisten, dengan fluktuasi yang hampir tidak ada. Kondisi ini menunjukkan bahwa jaringan di ruangan kuliah umum ini mampu berfungsi optimal bahkan jika digunakan untuk aplikasi yang memerlukan latensi rendah dan stabilitas tinggi, seperti video *streaming* berkualitas tinggi, atau simulasi *online*. Dengan stabilitas yang tinggi, aktivitas akademik dan kegiatan lain di ruangan ini dapat berjalan dengan lancar tanpa gangguan berarti.

#### **4.7 Perbandingan Waktu Respons *Ping* pada 3 September 2024 dan 5 September 2024 pada pukul 08:00 – 09:00**

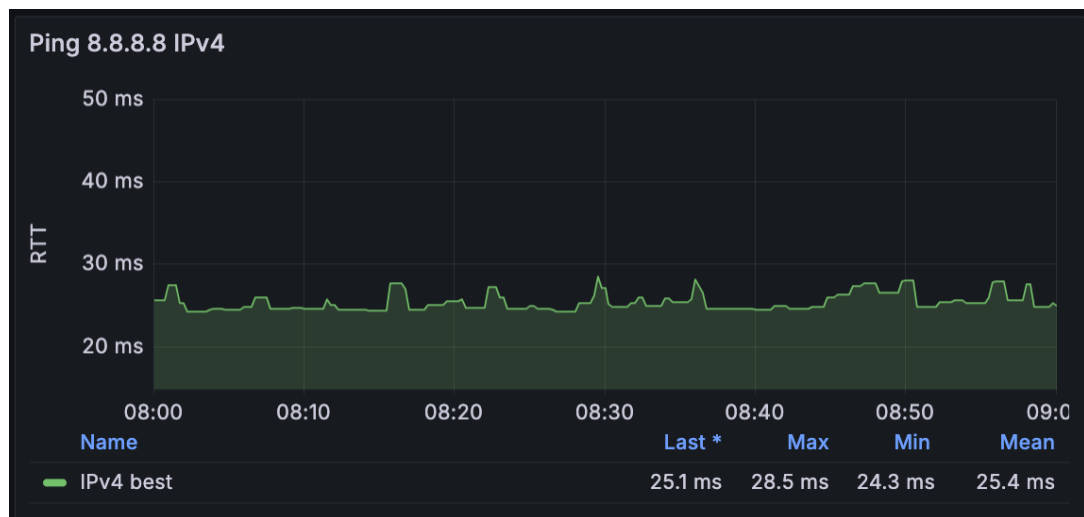
Dalam menganalisis performa jaringan, penting untuk melihat tidak hanya rata-rata waktu respons (*ping*), tetapi juga variabilitas yang terjadi pada waktu-waktu tertentu. Dua tanggal yang dianalisis dalam bab ini adalah 3 September 2024 dan 5 September 2024, yang mencerminkan aktivitas normal di Badan Sistem Informasi, khususnya di ruang developer. Setiap grafik memberikan gambaran mengenai waktu respons jaringan (*ping*) ke alamat IP 8.8.8.8 pada rentang waktu pukul 08:00 hingga 09:00.

Pada tanggal 3 September 2024, aktivitas di ruang developer adalah aktivitas kerja biasa yang dimulai dengan kedatangan pegawai. Sementara pada 5 September 2024, para pegawai mengikuti sesi *Zoom Meeting* untuk acara *TechTalk* yang diadakan secara rutin. Sekitar 27 hingga 30 orang di ruang developer terhubung ke *Zoom* menggunakan laptop mereka pada hari

ini, yang memberikan dampak signifikan pada performa jaringan. Perbandingan antara kedua hari ini akan memperlihatkan dampak aktivitas yang berbeda terhadap performa jaringan.

#### 4.7.1 Data Waktu Respons Ping

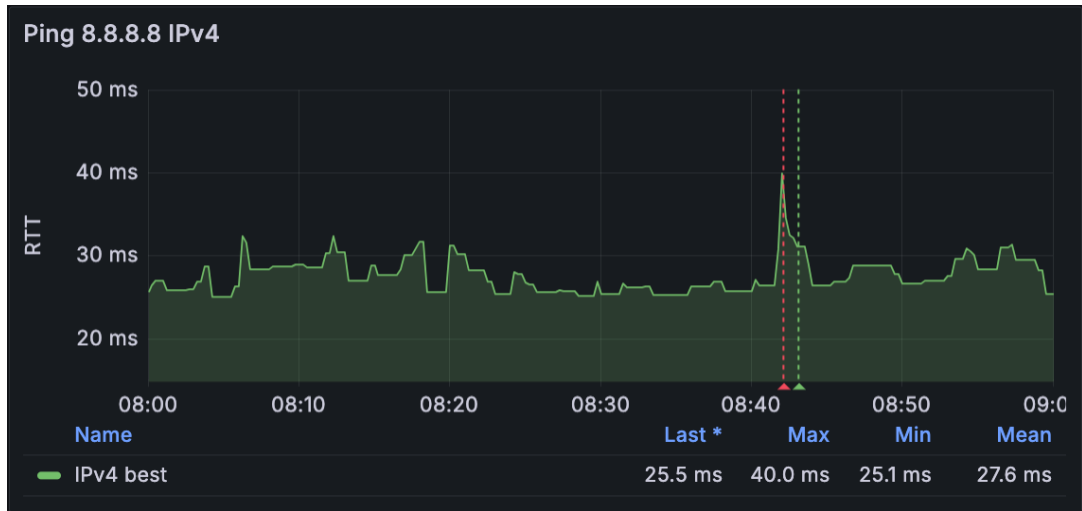
- a. Pada tanggal 3 September 2024, pada Gambar 4.3 waktu respons *ping* ke alamat IP 8.8.8.8 memiliki rata-rata 25.4 ms. Variasi yang terjadi pada waktu respons cukup kecil, dengan nilai minimum 24.3 ms dan maksimum 28.5 ms. Gambar 4.3 menunjukkan bahwa selama satu jam ini, tidak ada lonjakan yang signifikan, dan waktu respons tetap berada dalam batas wajar.



Gambar 4.3 Infografis *ping* pada tanggal 3 September 2024 pukul 08:00 – 09:00

- b. Pada tanggal 5 September 2024, seperti yang ditunjukkan pada Gambar 4.4, terjadi peningkatan rata-rata waktu respons *ping* menjadi 27,6 ms, dengan nilai minimum 25,1 ms dan nilai maksimum mencapai 40,0 ms. Peningkatan ini erat kaitannya dengan aktivitas *Zoom Meeting* yang berlangsung di ruang developer, di mana sekitar 27 hingga 30 orang terhubung secara bersamaan. Kondisi ini menciptakan beban tinggi pada jaringan, terutama karena penggunaan *Zoom*, yang memerlukan *bandwidth* yang cukup besar untuk menjaga kualitas audio dan video. *Zoom Meeting* yang dimulai pada pukul 08:00 membutuhkan alokasi *bandwidth* yang lebih tinggi dari biasanya, sehingga menghasilkan lonjakan waktu respons yang signifikan. Waktu respons yang mencapai 40,0 ms pada salah satu titik menjadi indikasi bahwa kapasitas jaringan sedang diuji dengan intensitas lalu lintas data yang tinggi.

Lonjakan ini menyoroti pentingnya pengelolaan *bandwidth* dan kapasitas jaringan yang optimal dalam menghadapi aktivitas berat seperti konferensi video. Selain itu, peningkatan waktu respons ini juga memicu sistem notifikasi otomatis seperti yang terlihat pada Gambar 4.5, di mana administrator jaringan menerima peringatan ketika metrik *ping* melebihi ambang batas yang telah ditetapkan. Pemberitahuan seperti pada ini memungkinkan administrator untuk segera mengevaluasi kondisi jaringan dan mempertimbangkan langkah-langkah mitigasi, seperti redistribusi *bandwidth* atau pembatasan akses bagi pengguna lain untuk menjaga stabilitas jaringan. Dengan sistem pemberitahuan yang tanggap, gangguan terhadap layanan jaringan dapat diminimalkan, sekaligus menjaga kelancaran aktivitas penting seperti *Zoom Meeting*.



Gambar 4.4 Infografis ping pada tanggal 5 September 2024 pukul 08:00 – 09:00

Panel: <http://localhost:3000/d/rYdddIPWk?orgId=1&viewPanel=330> September 5 06:53

paipal  
\*\*Firing\*\*

Value: D=40.00887680053711, E=1  
Labels:  
- alertname = Ping 8.8.8.8 IPv4  
- grafana\_folder = Alert  
- ip\_version = 4  
Annotations:  
Source:  
<http://localhost:3000/alerting/grafana/fdskskzdx883kd/view?orgId=1>  
Silence: [http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DPing+8.8.8.8+IPv4&matcher=grafana\\_folder%3DAlert&matcher=ip\\_version%3D4&orgId=1](http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DPing+8.8.8.8+IPv4&matcher=grafana_folder%3DAlert&matcher=ip_version%3D4&orgId=1)  
Dashboard: <http://localhost:3000/d/rYdddIPWk?orgId=1>  
Panel: <http://localhost:3000/d/rYdddIPWk?orgId=1&viewPanel=321> 08:42

\*\*Resolved\*\*

Value: D=31.19503402709961, E=0  
Labels:  
- alertname = Ping 8.8.8.8 IPv4  
- grafana\_folder = Alert  
- ip\_version = 4  
Annotations:  
Source:  
<http://localhost:3000/alerting/grafana/fdskskzdx883kd/view?orgId=1>  
Silence: [http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DPing+8.8.8.8+IPv4&matcher=grafana\\_folder%3DAlert&matcher=ip\\_version%3D4&orgId=1](http://localhost:3000/alerting/silence/new?alertmanager=grafana&matcher=alertname%3DPing+8.8.8.8+IPv4&matcher=grafana_folder%3DAlert&matcher=ip_version%3D4&orgId=1)  
Dashboard: <http://localhost:3000/d/rYdddIPWk?orgId=1>  
Panel: <http://localhost:3000/d/rYdddIPWk?orgId=1&viewPanel=321> 08:47

paipal  
\*\*Firing\*\*

Message

Gambar 4.5 Pemberitahuan ping saat melebihi ambang batas

#### 4.7.2 Analisis Performa Jaringan

- a. Pada 3 September 2024, waktu respons *ping* stabil sepanjang jam kerja, dengan rata-rata waktu respons yang tetap pada 25.4 ms. Grafik menunjukkan sedikit variasi pada waktu respons yang mencerminkan aktivitas normal pegawai yang masuk ke ruangan dan mulai bekerja, dengan peningkatan kecil saat mereka mengakses sistem dan aplikasi internal. Tidak ada aktivitas berat yang dilakukan pada hari ini, sehingga jaringan tetap dalam kondisi stabil. Peningkatan waktu respons yang terlihat pada sekitar pukul 08:10 dan 08:30 mungkin disebabkan oleh aktivitas akses jaringan awal, seperti *login* ke sistem, namun ini tidak memberikan dampak yang besar pada performa keseluruhan. Secara umum, fluktuasi ini dapat dianggap wajar dalam konteks aktivitas reguler pegawai.
- b. Pada 5 September 2024, waktu respons meningkat secara signifikan dibandingkan dengan 3 September. Rata-rata waktu respons mencapai 27.6 ms, dengan lonjakan hingga 40.0 ms. Lonjakan ini terjadi karena sekitar 27 hingga 30 pegawai terhubung ke *Zoom Meeting* untuk acara *TechTalk* yang dimulai pada pukul 08:00. Aktivitas konferensi video membutuhkan *bandwidth* yang lebih besar, yang menyebabkan waktu respons jaringan melambat. Puncak waktu respons yang terjadi pada pukul 08:40 menunjukkan beban tertinggi pada jaringan, yang disebabkan oleh acara *TechTalk* tersebut. Meskipun demikian, setelah lonjakan ini, waktu respons kembali stabil di sekitar 25.5 ms hingga 25.6 ms, menunjukkan bahwa setelah penggunaan *bandwidth* yang intensif pada awal sesi, jaringan mampu kembali ke performa yang lebih baik.

#### 4.7.3 Interpretasi Hasil *Ping* pada 3 September 2024 dan 5 September 2024 pada pukul 08:00 – 09:00

Dari perbandingan kedua tanggal, terlihat bahwa aktivitas rutin pegawai di ruang developer pada 3 September 2024 menyebabkan waktu respons *ping* yang stabil dan tidak menunjukkan lonjakan yang signifikan. Dengan rata-rata waktu respons 25.4 ms, performa jaringan pada hari tersebut berada dalam kondisi optimal untuk aktivitas sehari-hari seperti *login* ke sistem atau akses aplikasi internal.

Sebaliknya, pada 5 September 2024, aktivitas yang lebih berat, seperti *Zoom Meeting*, menyebabkan peningkatan waktu respons *ping* hingga 40.0 ms, dengan rata-rata 27.6 ms. Sekitar 27 hingga 30 orang yang terhubung ke *Zoom* secara bersamaan memberikan tekanan tambahan pada jaringan. Meskipun terjadi lonjakan signifikan pada awal sesi *Zoom*, jaringan

mampu kembali stabil setelah beberapa waktu, menunjukkan bahwa jaringan tetap mampu menangani beban berat meski ada peningkatan latensi.

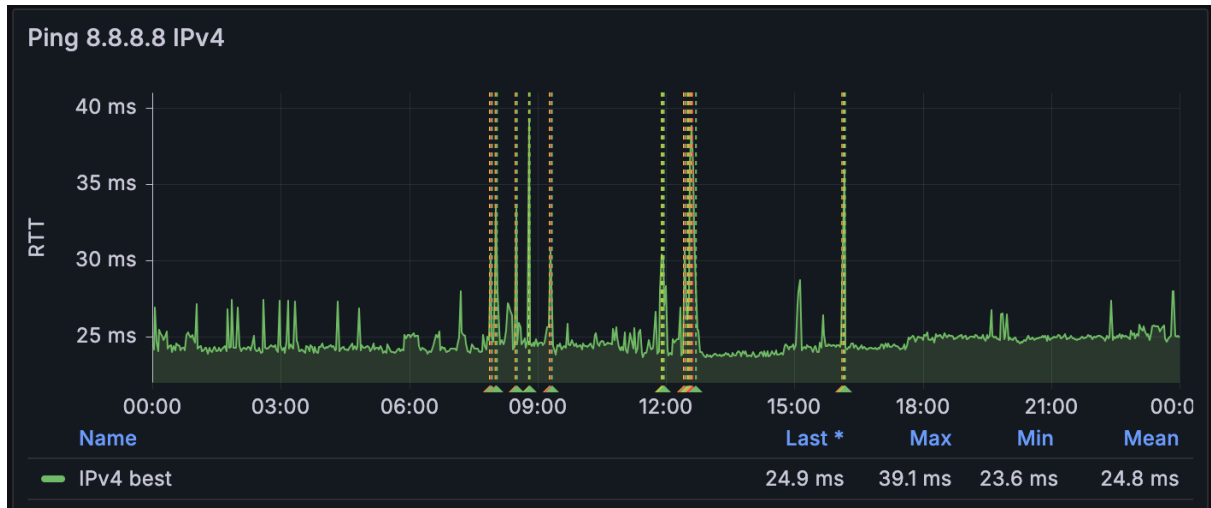
Secara keseluruhan, performa jaringan di ruang developer menunjukkan stabilitas yang baik pada hari-hari normal, namun aktivitas berat seperti konferensi video dapat memberikan tekanan pada jaringan, yang menyebabkan peningkatan waktu respons yang perlu diantisipasi oleh tim pengelola jaringan.

#### **4.8 Perbandingan Waktu Respons Ping pada 12 hingga 15 November 2024 di Ruangan Kuliah Umum Dr. Sardjito UII**

Dalam menganalisis performa jaringan di ruangan kuliah umum yang terletak di lantai 1 Gedung Kuliah Umum Dr. Sardjito Universitas Islam Indonesia, penting untuk memahami pola waktu respons *ping* serta variabilitasnya. Ruangan ini memiliki kapasitas hingga 150 orang, sehingga aktivitas penggunaan jaringan oleh sejumlah besar perangkat dapat memengaruhi performa koneksi internet. Analisis dilakukan terhadap waktu respons *ping* ke server Google Public DNS (8.8.8.8) selama empat hari berturut-turut, yaitu pada tanggal 12, 13, 14, dan 15 November 2024, untuk mengidentifikasi stabilitas jaringan di lokasi tersebut.

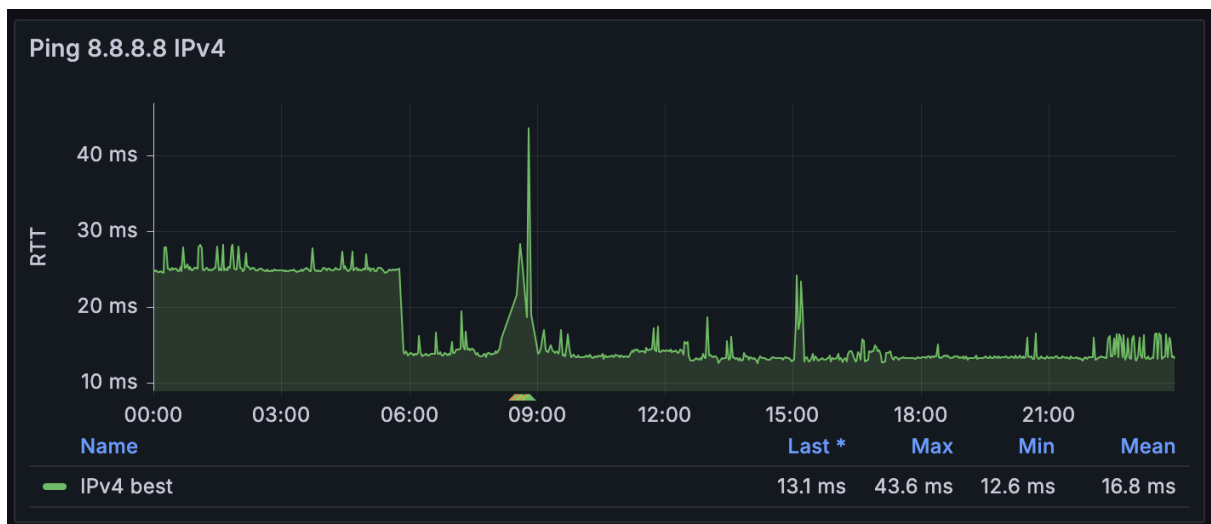
##### **4.8.1 Data Waktu Respon *Ping***

- a. Pada 12 November 2024, rata-rata waktu respons *ping* tercatat sebesar 24.8 ms dengan nilai rata-rata deviasi 0.67 ms dan standar deviasi 0.78 ms. Grafik *ping* menunjukkan fluktuasi yang sangat kecil, menandakan jaringan dalam kondisi sangat stabil. Variasi waktu respons *ping* berada dalam rentang yang minimal tanpa lonjakan signifikan, seperti terlihat pada Gambar 4.6. Kondisi ini menunjukkan jaringan mampu menangani aktivitas rutin dengan baik, memberikan pengalaman penggunaan yang optimal.



Gambar 4.6 Infografis *ping* pada tanggal 12 November 2024

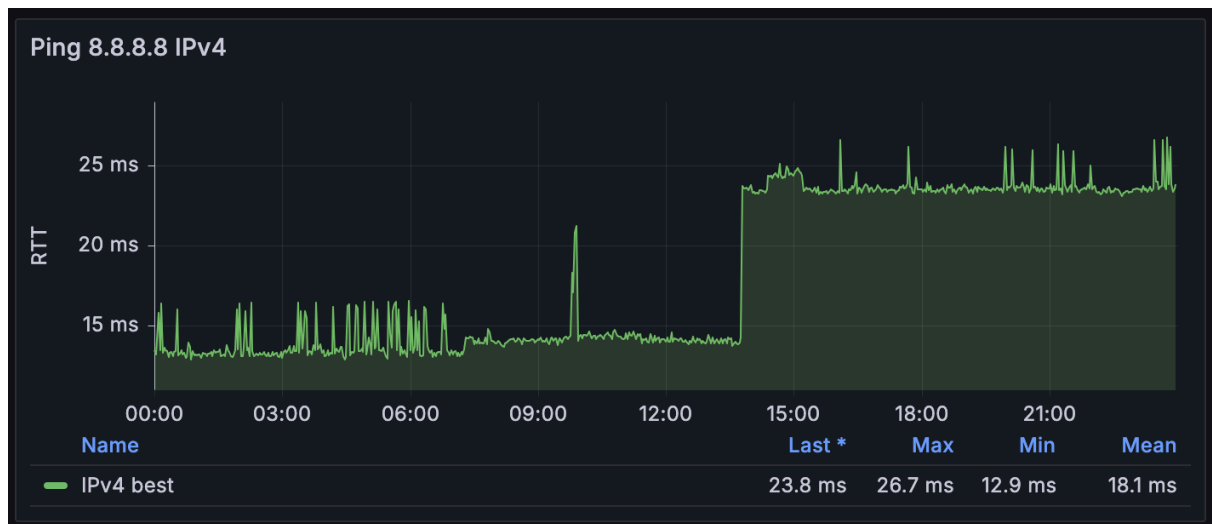
- b. Pada 13 November 2024, rata-rata waktu respons tetap 25.05 ms, namun dengan nilai rata-rata deviasi yang meningkat menjadi 4.44 ms dan standar deviasi 4.76 ms. Grafik menunjukkan fluktuasi yang lebih besar dibandingkan tanggal lainnya, dengan beberapa lonjakan waktu respons yang signifikan. Ketidakstabilan ini dapat disebabkan oleh tingginya beban penggunaan jaringan pada waktu tertentu, seperti saat banyak perangkat melakukan aktivitas berat secara bersamaan. Gambar 4.7 memperlihatkan bahwa stabilitas jaringan terganggu, dengan fluktuasi yang jauh lebih besar dibandingkan hari sebelumnya.



Gambar 4.7 Infografis *ping* pada tanggal 13 November 2024

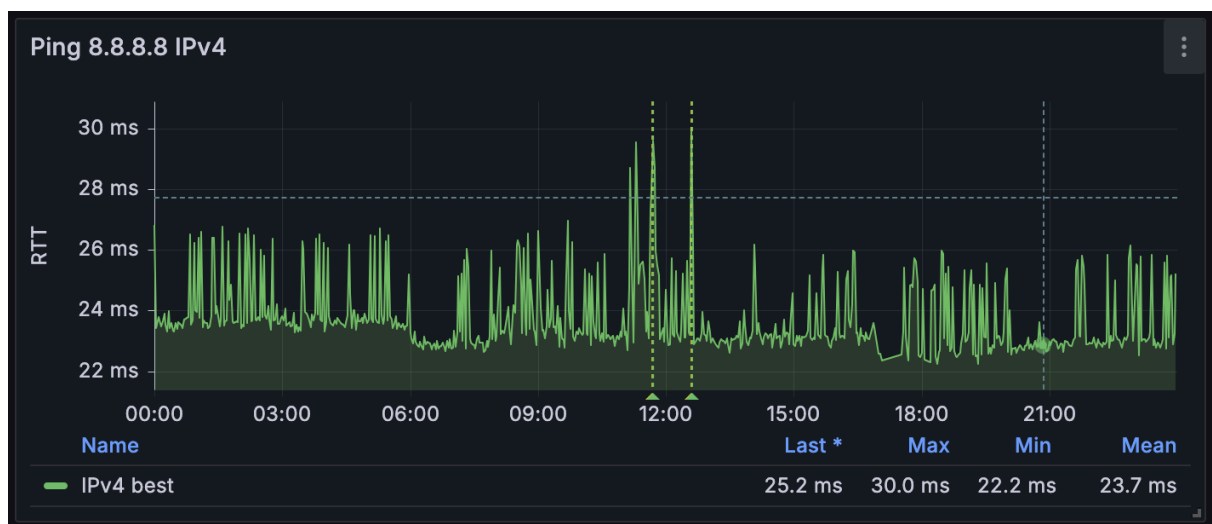
- c. Pada 14 November 2024, rata-rata waktu respons lebih rendah, yaitu 16.92 ms, namun nilai rata-rata deviasi 3.72 ms dan standar deviasi 4.01 ms menunjukkan adanya fluktuasi yang cukup signifikan. Grafik ping pada Gambar 4.8 memperlihatkan adanya beberapa lonjakan pada waktu tertentu, meskipun secara keseluruhan waktu respons masih cukup cepat.

Ketidakstabilan jaringan pada tanggal ini mungkin disebabkan oleh aktivitas yang lebih intens di periode tertentu.



Gambar 4.8 Infografis *ping* pada tanggal 14 November 2024

- d. Pada 15 November 2024, jaringan kembali menunjukkan performa yang sangat stabil, dengan rata-rata waktu respons 24.02 ms, rata-rata deviasi 0.30 ms, dan standar deviasi 0.40 ms. Grafik ping Gambar 4.9 menunjukkan fluktuasi yang minimal, tanpa adanya lonjakan signifikan. Kondisi ini mencerminkan stabilitas jaringan yang optimal, cocok untuk berbagai aktivitas yang membutuhkan koneksi internet yang stabil.



Gambar 4.9 Infografis *ping* pada tanggal 15 November 2024

#### 4.8.2 Analisis Performa Jaringan

- a. Pada tanggal 12 November 2024, performa jaringan pada tanggal ini stabil dengan fluktuasi minimal. Aktivitas di ruangan tidak memberikan tekanan berarti pada jaringan, sehingga

waktu respons tetap konsisten. Kondisi ini ideal untuk aktivitas rutin seperti *browsing*, akses LMS, atau *streaming* video berkualitas tinggi.

- b. Pada tanggal 13 November 2024 menunjukkan performa jaringan yang paling tidak stabil di antara empat tanggal yang dianalisis. Fluktuasi besar dapat berdampak pada pengalaman pengguna, terutama pada aktivitas sensitif terhadap latensi seperti video *streaming* atau presentasi daring. Hal ini mungkin disebabkan oleh penggunaan perangkat secara simultan oleh banyak pengguna.
- c. Pada tanggal 14 November 2024, meskipun waktu respons rata-rata lebih rendah, fluktuasi yang signifikan menunjukkan adanya gangguan stabilitas jaringan di beberapa waktu. Kondisi ini masih dapat diterima untuk aktivitas reguler, tetapi bisa memberikan dampak pada aplikasi *real-time* yang membutuhkan koneksi stabil.
- d. Pada tanggal 15 November 2024, jaringan kembali sangat stabil, dengan fluktuasi minimal. Kondisi ini memberikan pengalaman pengguna yang optimal untuk semua jenis aktivitas, baik reguler maupun yang membutuhkan latensi rendah seperti konferensi video.

#### **4.8.3 Interpretasi Hasil Performa Jaringan pada 12 hingga 15 November 2024 di Ruang Kuliah Umum Dr. Sardjito UII**

Dari perbandingan empat tanggal, terlihat bahwa:

- a. Jaringan paling stabil: 12 dan 15 November 2024, dengan rata-rata deviasi dan standar deviasi yang rendah.
- b. Jaringan paling tidak stabil: 13 November 2024, dengan nilai rata-rata deviasi dan standar deviasi tertinggi, menunjukkan fluktuasi yang besar.
- c. Jaringan dengan waktu respons tercepat: 14 November 2024, meskipun fluktuasi yang tercatat cukup besar.

Secara keseluruhan, performa jaringan di ruangan kuliah umum ini menunjukkan kemampuan untuk mendukung aktivitas akademik, meskipun ada hari tertentu dengan peningkatan variabilitas jaringan yang perlu diantisipasi. Optimalisasi jaringan dan pengelolaan *bandwidth* yang baik dapat membantu menjaga stabilitas jaringan untuk mendukung kegiatan di ruangan dengan kapasitas besar seperti ini.

#### 4.9 Perbandingan Performa Jaringan Ruangan Developer BSI UII dan Ruang Kuliah Lantai 1 Gedung Kuliah Umum Dr. Sardjito UII

Dalam analisis ini, dilakukan perbandingan antara performa jaringan di Ruangan Developer BSI UII, yang menggunakan *Access Point* Ruckus R650, dengan Ruang Kuliah Lt. 1 Gedung Kuliah Umum Dr. Sardjito UII, yang menggunakan *Access Point* Ruckus R750. Kedua ruangan memiliki karakteristik penggunaan jaringan yang berbeda, baik dari sisi jumlah pengguna maupun intensitas aktivitas jaringan. Perbandingan ini memberikan wawasan tentang bagaimana perangkat keras jaringan berperforma di lingkungan yang berbeda.

##### 4.9.1 Ruang Developer BSI UII

Ruangan ini dilengkapi dengan *Access Point* Ruckus R650, yang dirancang untuk menangani hingga 512 perangkat secara simultan. Pada tanggal 3 dan 5 September 2024, jumlah pengguna aktif di ruangan ini berkisar antara 27 hingga 30 orang, dengan aktivitas yang melibatkan akses aplikasi internal dan konferensi video menggunakan *Zoom*.

Tabel 4.8 Kinerja jaringan pada tanggal 3 September 2024

Rata-rata <i>ping</i>	Rata-rata deviasi	Standar deviasi	Stabilitas jaringan
25.41 <i>ms.</i>	0.273 <i>ms.</i>	0.318 <i>ms.</i>	Sangat tinggi, tanpa lonjakan signifikan, mencerminkan performa yang optimal untuk aktivitas rutin seperti <i>login</i> ke sistem dan akses aplikasi internal.

Tabel 4.9 Kinerja jaringan pada tanggal 5 September 2024

Rata-rata <i>ping</i>	Rata-rata deviasi	Standar deviasi	Stabilitas jaringan
25.76 <i>ms.</i>	0.527 <i>ms.</i>	0.784 <i>ms.</i>	Masih cukup tinggi, meskipun terjadi peningkatan waktu respons hingga 27.6 <i>ms</i> saat 27-30 pengguna terhubung ke <i>Zoom Meeting</i> secara bersamaan. Beban tambahan dari konferensi video memberikan sedikit tekanan pada

			jaringan, namun tidak sampai mengganggu stabilitas secara signifikan.
--	--	--	-----------------------------------------------------------------------

#### 4.9.2 Ruang Kuliah Lantai 1 Gedung Kuliah Umum Dr. Sardjito

Ruang kuliah ini menggunakan *Access Point* Ruckus R750, yang memiliki kemampuan menangani hingga 1.024 perangkat secara simultan. Dengan kapasitas ruangan hingga 150 orang, perangkat ini dirancang untuk lingkungan dengan tingkat aktivitas jaringan yang tinggi. Pengujian dilakukan selama 12 hingga 15 November 2024.

Tabel 4.10 Kinerja jaringan pada tanggal 12 November 2024

Rata-rata <i>ping</i>	Rata-rata deviasi	Standar deviasi	Stabilitas jaringan
25.05 <i>ms</i>	0.67 <i>ms</i> .	0.78 <i>ms</i> .	Sangat baik, dengan fluktuasi minimal, cocok untuk aktivitas perkuliahan seperti akses <i>Learning Management System</i> (LMS) dan konferensi video.

Tabel 4.11 Kinerja jaringan pada tanggal 13 November 2024

Rata-rata <i>ping</i>	Rata-rata deviasi	Standar deviasi	Stabilitas jaringan
25.05 <i>ms</i>	4.44 <i>ms</i> .	4.76 <i>ms</i> .	Cenderung menurun, dengan fluktuasi signifikan akibat banyaknya perangkat yang terhubung secara bersamaan. Lonjakan latensi dapat memengaruhi pengalaman pengguna dalam aktivitas sensitif latensi seperti <i>streaming</i> video <i>real-time</i> .

Tabel 4.12 Kinerja jaringan pada tanggal 14 November 2024

Rata-rata <i>ping</i>	Rata-rata deviasi	Standar deviasi	Stabilitas jaringan
-----------------------	-------------------	-----------------	---------------------

16.92 ms	3.72 ms.	4.01 ms.	Fluktuasi yang cukup besar terlihat, meskipun waktu respons rata-rata lebih rendah. Ketidakstabilan ini kemungkinan besar disebabkan oleh aktivitas pengguna yang lebih intens pada waktu tertentu.
----------	----------	----------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabel 4.13 Kinerja jaringan pada tanggal 15 November 2024

Rata-rata ping	Rata-rata deviasi	Standar deviasi	Stabilitas jaringan
24.02 ms	0.30 ms.	0.40 ms.	Kembali sangat baik, dengan fluktuasi minimal, mencerminkan jaringan yang stabil untuk mendukung berbagai aktivitas akademik.

### 4.9.3 Analisis Perbandingan

*Access Point* Ruckus R650 yang digunakan di Ruang Developer BSI UII terbukti cocok untuk lingkungan dengan jumlah perangkat moderat, hingga 30 pengguna aktif. Perangkat ini menunjukkan performa yang sangat stabil, bahkan ketika terjadi peningkatan aktivitas berat seperti saat *Zoom Meeting*. Fluktuasi waktu respons yang sangat kecil mengindikasikan bahwa R650 mampu menangani aktivitas rutin dengan efisiensi tinggi, menjadikannya solusi ideal untuk ruang kerja yang membutuhkan konektivitas konsisten dan andal.

Sementara itu, *Access Point* Ruckus R750 yang digunakan di Ruang Kuliah Lt. 1 Gedung Kuliah Umum Dr. Sardjito dirancang untuk menangani jumlah perangkat yang jauh lebih besar, hingga 150 pengguna aktif. Performa perangkat ini cukup baik dalam kondisi penggunaan normal, namun pada tanggal dengan aktivitas tinggi, seperti 13 November, terlihat adanya fluktuasi signifikan yang menunjukkan tekanan pada jaringan akibat intensitas penggunaan yang tinggi. Meskipun demikian, stabilitas jaringan kembali optimal pada tanggal dengan beban pengguna yang lebih merata, seperti 12 dan 15 November, menjadikannya pilihan tepat untuk ruang kuliah yang memerlukan konektivitas andal untuk mendukung berbagai aktivitas simultan.

#### 4.9.4 Kesimpulan Perbandingan Performa Jaringan Ruangan Developer BSI UII dan Ruang Kuliah Lt. 1 Gedung Kuliah Umum Dr. Sardjito UII

*Access Point* Ruckus R650 ideal digunakan di ruangan dengan kapasitas pengguna sedang, seperti ruangan developer, yang membutuhkan koneksi stabil untuk aktivitas berat sesekali. Perangkat ini memberikan performa optimal untuk aplikasi rutin seperti akses aplikasi internal atau konferensi video dengan jumlah pengguna yang terbatas. Sementara itu, *Access Point* Ruckus R750 sangat cocok untuk ruangan dengan kapasitas besar, seperti ruang kuliah umum, yang melibatkan aktivitas simultan oleh banyak perangkat. Namun, untuk memastikan kinerja optimal, pengelolaan *bandwidth* yang baik diperlukan, terutama pada hari-hari dengan lonjakan penggunaan, seperti ketika banyak pengguna terhubung secara bersamaan.

Untuk mengoptimalkan jaringan, di ruangan developer, aktivitas berat pada perangkat non-prioritas sebaiknya dibatasi selama sesi penting seperti *Zoom Meeting* untuk menjaga stabilitas jaringan. Di ruang kuliah, pembagian kanal jaringan atau penerapan kebijakan penggunaan *bandwidth* dapat membantu mengurangi fluktuasi pada hari dengan aktivitas tinggi. Dengan memahami kebutuhan spesifik dari kedua ruangan ini, pengelolaan jaringan dapat dioptimalkan untuk memberikan pengalaman pengguna yang konsisten dan memuaskan.

#### 4.10 Pengambilan Data di Ruang Teatrikal Gedung Dr. Sardjito UII

Berdasarkan hasil pengambilan data jaringan di ruang teatrikal Gedung Dr. Sardjito, Universitas Islam Indonesia, pada tanggal 24 September 2024, dilakukan analisis terhadap dua metrik utama, yaitu *ping* dan *packet loss*. Hasil ini memberikan gambaran mengenai performa jaringan selama periode pemantauan, terutama ketika terjadi lonjakan pengguna atau perubahan beban jaringan.

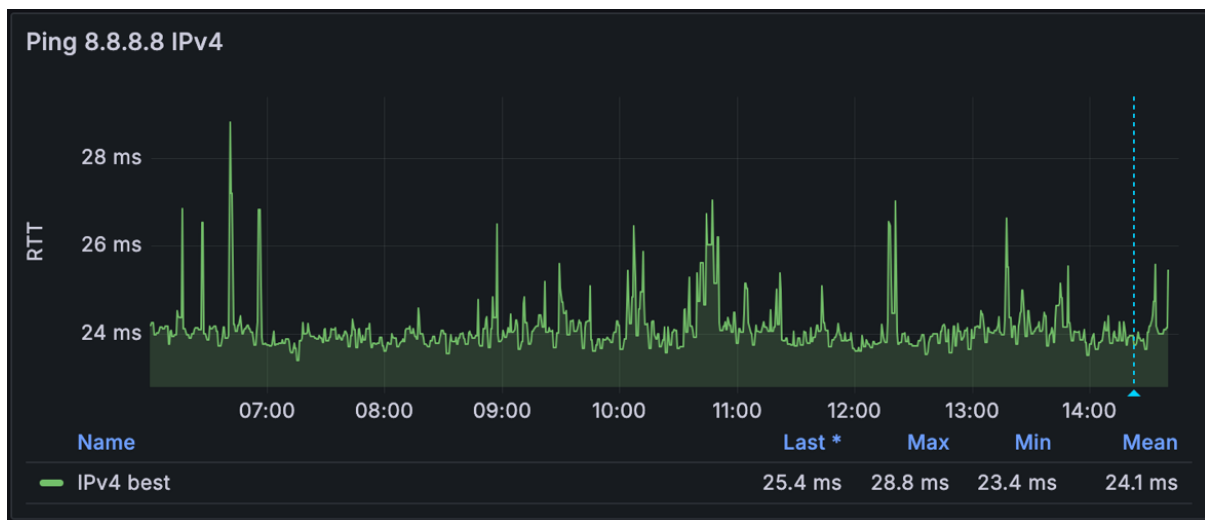
##### 4.10.1 *Ping* di Ruang Teatrikal Gedung Dr. Sardjito UII

Pada pengujian *ping*, seperti yang ditunjukkan pada Gambar 4.10, rata-rata waktu respons jaringan (*Round-Trip Time* atau *RTT*) berada pada 24.1 ms, dengan waktu respons maksimum mencapai 28.8 ms. Secara keseluruhan, performa jaringan dapat dikategorikan stabil selama periode pemantauan, meskipun terdapat beberapa lonjakan pada waktu-waktu tertentu, terutama sekitar pukul 06:00 dan 10:30.

Lonjakan-lonjakan tersebut dapat mengindikasikan adanya peningkatan beban jaringan, yang kemungkinan besar disebabkan oleh peningkatan jumlah perangkat yang terhubung atau aktivitas yang lebih intens. Fluktuasi kecil yang terlihat pada grafik *ping* ini dapat berdampak

pada aplikasi yang sensitif terhadap latensi, seperti panggilan video atau permainan daring, meskipun secara umum performa jaringan masih dalam batas wajar untuk aktivitas sehari-hari.

Berdasarkan hasil analisis *packet loss* yang diambil dari ruang teatrikal gedung Dr. Sardjito Universitas Islam Indonesia pada tanggal 24 September 2024, dapat disimpulkan bahwa performa jaringan menunjukkan adanya fluktuasi dalam tingkat *packet loss* selama periode pemantauan.

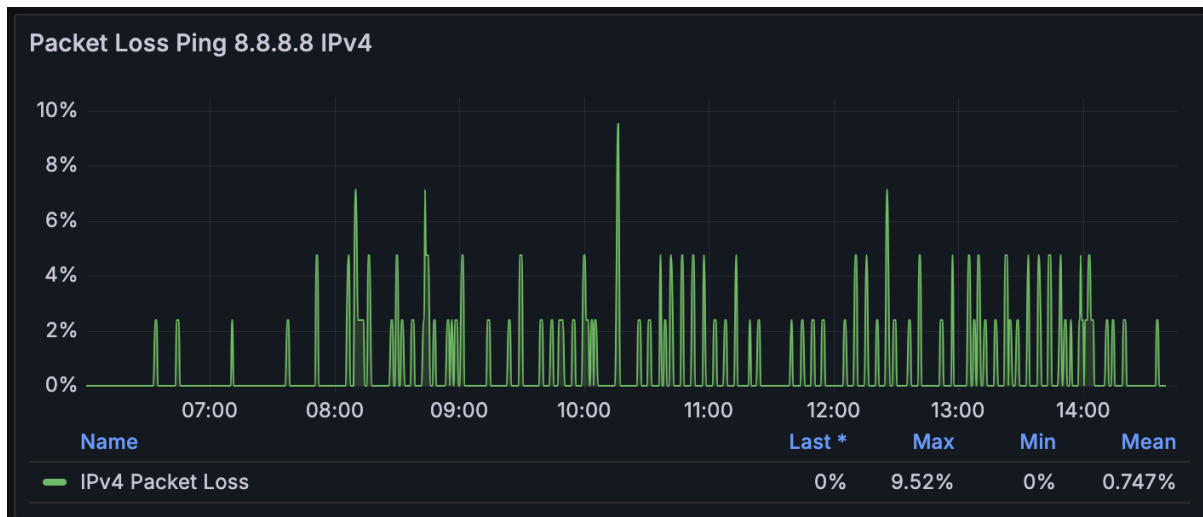


Gambar 4.10 Infografis *ping* di Ruang Teatrikal Gedung Dr. Sardjito UII

#### 4.10.2 *Packet loss* di Ruang Teatrikal Gedung Dr. Sardjito UII

Selain *ping*, analisis *packet loss* juga memberikan indikasi penting mengenai stabilitas jaringan. Gambar 4.11 menampilkan tingkat *packet loss* selama periode pengujian, dengan variasi antara 0% hingga 9.52%. Rata-rata *packet loss* yang tercatat adalah 0.747%, menunjukkan bahwa meskipun *packet loss* relatif rendah secara keseluruhan, terdapat beberapa lonjakan signifikan yang terjadi, terutama sekitar pukul 08:00, 10:30, dan 12:30.

Tingkat *packet loss* yang mencapai lebih dari 5% pada waktu-waktu tersebut menunjukkan adanya gangguan sementara pada jaringan, yang bisa diakibatkan oleh peningkatan beban atau masalah sementara pada jaringan. Lonjakan *packet loss* ini dapat berdampak langsung pada aplikasi yang membutuhkan transfer data yang stabil dan konsisten, seperti video *streaming* atau komunikasi *real-time*, karena hilangnya paket data dapat menyebabkan penurunan kualitas layanan.



Gambar 4.11 Infografis *packet loss* di Ruang Teatrikal Gedung Dr. Sardjito UII

#### 4.11 Pengambilan Data di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII

Pada tanggal 28 September 2024, dilakukan pengambilan data jaringan di Auditorium Prof. K.H. Abdul Kahar Mudzakkir Universitas Islam Indonesia, bersamaan dengan pelaksanaan wisuda hari pertama periode pertama tahun ajaran 2024/2025. Acara tersebut dihadiri oleh sekitar 500 orang, yang terdiri dari peserta wisuda dan panitia. Kepadatan peserta di auditorium ini menyebabkan tingginya beban jaringan yang berdampak pada kualitas koneksi selama periode pengambilan data. Analisis dilakukan terhadap dua metrik utama, yaitu *ping* dan *packet loss*, untuk mengevaluasi performa jaringan dalam kondisi beban yang tinggi.

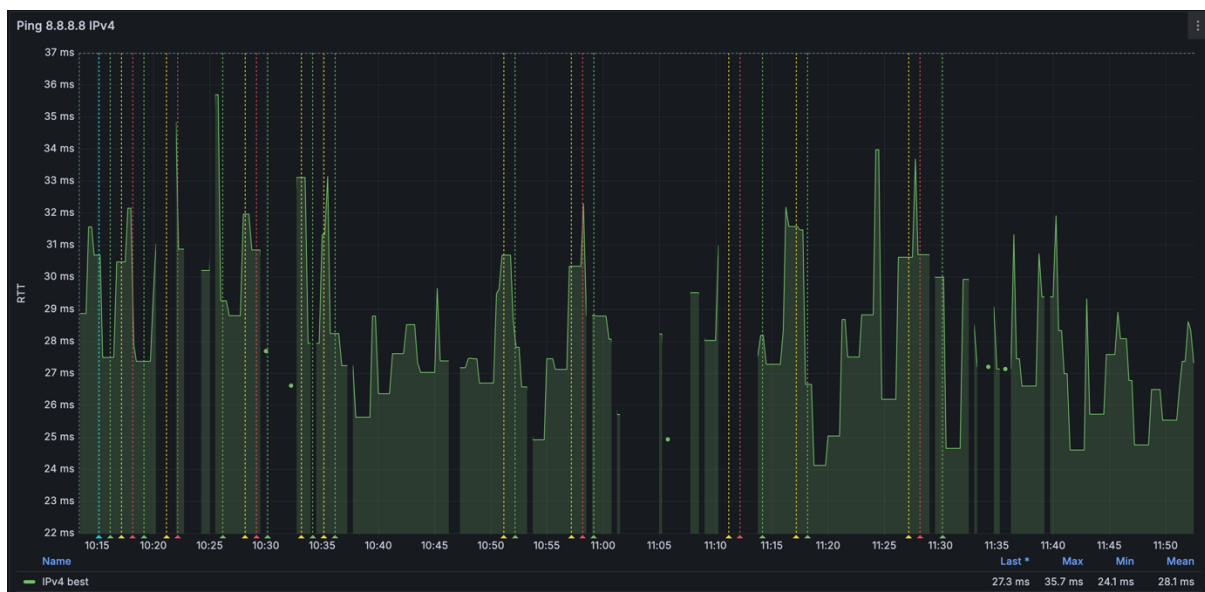
##### 4.11.1 *Ping* di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII

Pada pengujian *ping* yang dilakukan di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII, seperti yang terlihat pada Gambar 4.12, diperoleh hasil waktu respons rata-rata (*Round-Trip Time* atau RTT) sebesar 24,1 ms, dengan *ping* maksimum mencapai 28,8 ms. Secara umum, performa jaringan di lokasi ini cukup stabil, namun terdapat beberapa lonjakan signifikan yang terjadi pada rentang waktu tertentu, terutama sekitar pukul 10:25 dan 11:30. Lonjakan ini menunjukkan adanya peningkatan latensi yang kemungkinan besar diakibatkan oleh peningkatan jumlah perangkat yang terhubung secara bersamaan ke *access point* di lokasi tersebut.

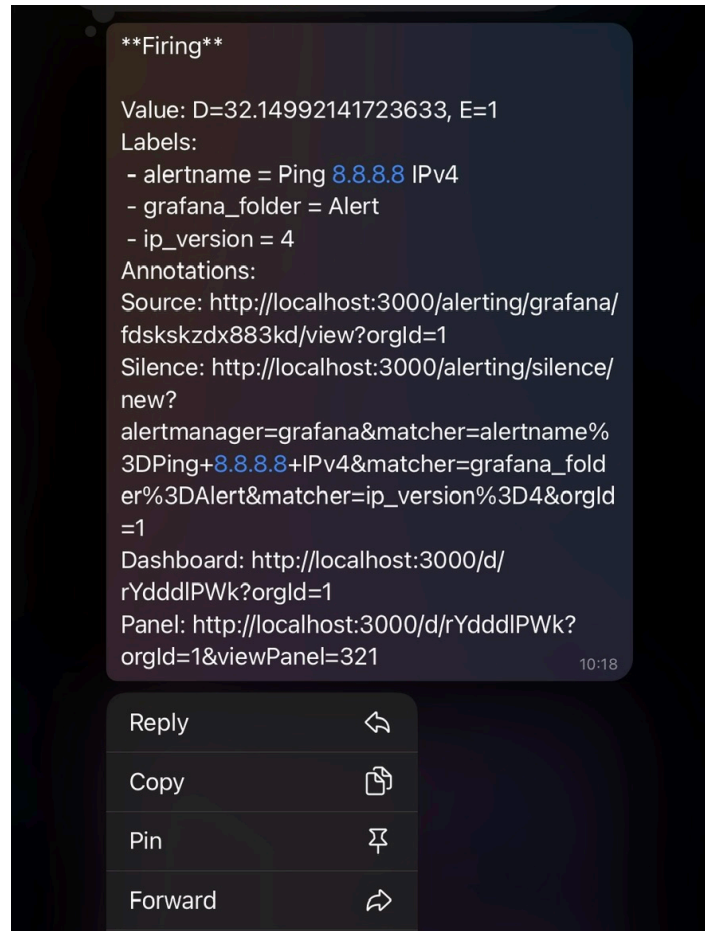
Grafik *ping* memperlihatkan fluktuasi yang terjadi selama acara berlangsung, yang menggambarkan adanya perubahan kondisi jaringan ketika jumlah pengguna meningkat secara drastis. Waktu respons yang meningkat tajam pada beberapa titik waktu menunjukkan bahwa

jaringan mengalami beban yang signifikan, yang dapat mempengaruhi kualitas koneksi bagi pengguna. Kondisi ini terutama berdampak pada aplikasi-aplikasi yang sensitif terhadap latensi, seperti panggilan video dan *streaming*. Apabila tidak ditangani dengan baik, peningkatan latensi ini dapat mengganggu pengalaman pengguna yang membutuhkan koneksi stabil dan cepat.

Sebagai langkah mitigasi, sistem *edge sensors* yang digunakan memberikan notifikasi secara otomatis saat waktu respons *ping* melebihi ambang batas yang telah ditentukan, seperti yang ditunjukkan pada Gambar 4.13. Pemberitahuan ini memungkinkan administrator jaringan untuk segera mengambil tindakan korektif, seperti melakukan redistribusi *bandwidth* atau mengelola prioritas akses pengguna, untuk memastikan kualitas koneksi tetap terjaga meskipun beban jaringan meningkat. Dengan pemantauan yang konsisten dan pemberitahuan yang responsif, gangguan pada layanan jaringan dapat diminimalkan, sehingga memastikan acara atau aktivitas yang berlangsung di auditorium dapat berjalan dengan lancar tanpa hambatan berarti.



Gambar 4.12 Infografis *ping* di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII



Gambar 4.13 Pemberitahuan *ping* saat melebihi ambang batas

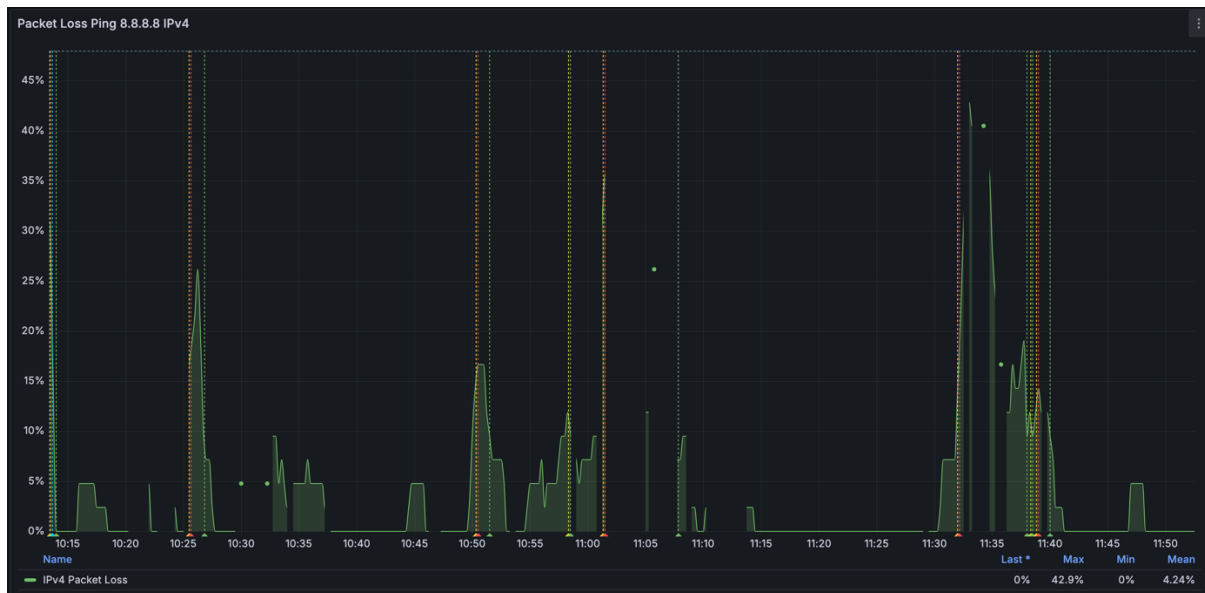
#### 4.11.2 *Packetloss* di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII

Selain pengujian *ping*, analisis *packet loss* juga memberikan wawasan penting mengenai kestabilan jaringan selama acara wisuda. Gambar 4.14 menunjukkan variasi tingkat *packet loss* selama periode pengambilan data, dengan hasil yang bervariasi antara 0% hingga 42,9%, dan rata-rata sebesar 4,24%. Lonjakan tertinggi terjadi sekitar pukul 10:25 dan 11:30, yang secara bersamaan juga tercatat sebagai waktu terjadinya lonjakan dalam metrik *ping*. Tingginya tingkat *packet loss* pada waktu-waktu tertentu menjadi indikasi adanya gangguan serius pada proses transmisi data, yang kemungkinan besar disebabkan oleh banyaknya perangkat yang terhubung secara bersamaan hingga melampaui kapasitas jaringan.

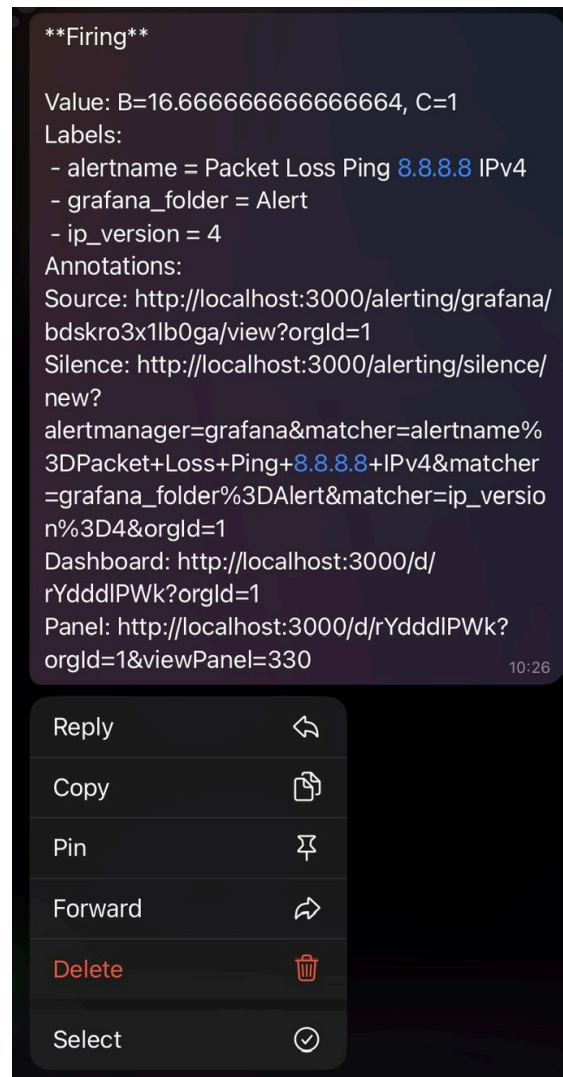
*Packet loss* yang tinggi dapat berdampak langsung pada stabilitas dan kualitas koneksi jaringan, terutama untuk aplikasi yang membutuhkan transfer data secara real-time, seperti *video streaming* atau konferensi video. Hilangnya paket data ini dapat mengakibatkan jeda, penurunan kualitas video, hingga terputusnya koneksi, yang tentu saja memengaruhi

pengalaman pengguna secara negatif, terutama pada saat-saat krusial selama acara wisuda berlangsung.

Untuk menangani masalah ini, sistem *edge sensors* yang digunakan memberikan notifikasi secara otomatis saat tingkat *packet loss* melebihi ambang batas yang telah ditentukan, seperti yang terlihat pada Gambar 4.15. Pemberitahuan ini memungkinkan tim IT atau administrator jaringan untuk segera mengambil langkah-langkah korektif yang diperlukan. Misalnya, mereka dapat menyesuaikan konfigurasi jaringan, mengalokasikan kembali *bandwidth*, atau membatasi jumlah perangkat yang terhubung. Dengan pendekatan yang proaktif melalui notifikasi ini, gangguan pada kualitas jaringan dapat diminimalkan, sehingga acara wisuda tetap berjalan dengan lancar tanpa hambatan besar pada konektivitas yang diperlukan untuk aplikasi-aplikasi penting.



Gambar 4.14 Infografis *packet loss* di Auditorium Prof. K.H. Abdul Kahar Mudzakkir UII



Gambar 4.15 Pemberitahuan *packet loss* saat melebihi ambang batas

#### 4.12 Mengatur Bot Notifikasi

Untuk mengatur notifikasi ke Telegram menggunakan *alert* di Grafana, langkah pertama adalah membuat bot Telegram. Buka aplikasi Telegram dan cari bot @BotFather, lalu buat bot baru dengan mengirimkan perintah /newbot dan ikuti instruksi untuk mendapatkan token API bot. Kemudian, buat *channel* baru di Telegram dan tambahkan bot yang baru saja dibuat sebagai admin di *channel* tersebut. Setelah itu, masuk ke dasbor Grafana dan buka pengaturan (*Configuration*), lalu pilih *Alerting*. Tambahkan *channel* notifikasi baru dengan memilih Telegram, kemudian masukkan token API bot yang telah didapatkan dan ID *chat/channel* Telegram. Untuk membuat *alert*, buka panel yang menampilkan data *ping* yang ingin diawasi. Tambahkan *rule alert* dengan mengklik *Alert* dan pilih *Create Alert*. Masukkan *query* yang telah disesuaikan untuk *alert*, misalnya seperti pada Gambar 4.16:

```
avg by(ip_version) (ping_rtt_best_seconds{job=~"pi4 ping_exporter",
instance=~"100.125.16.50:9427", target=~"8\\.8\\.8\\.8", ip_version=~"(4|6)"} *
1000
```

Gambar 4.16 *Query ping 8.8.8.8* pada Grafana

## 4.13 Pengaturan *Alert* Sistem Monitoring di Grafana Berdasarkan Data Variabilitas

### *Ping*

Pengaturan *alert* di Grafana bertujuan untuk memantau waktu respons *ping* jaringan secara *real-time* dan memberikan notifikasi jika terjadi gangguan yang signifikan. Berdasarkan analisis data variabilitas *ping*, pengaturan ini disederhanakan tanpa membedakan level *alert* seperti *warning* atau *critical*, dengan fokus pada satu kondisi ambang batas.

#### 4.13.1 Analisis Data untuk Penentuan Ambang Batas

Dari analisis statistik, rata-rata waktu respons *ping* pada kondisi normal berada di kisaran 25–26 *ms*, dengan fluktuasi yang minimal pada hari-hari stabil (3 September 2024 dan 15 November 2024). Peningkatan waktu respons hingga 40 *ms* tercatat pada saat beban tinggi (13 November 2024).

Untuk pengaturan *alert*, dipilih ambang batas waktu respons 30 *ms*, yang berada di atas rata-rata waktu respons normal tetapi cukup sensitif untuk mendeteksi potensi gangguan jaringan.

#### 4.13.2 Pengaturan Ambang Batas dan Durasi Pengamatan

Pengaturan *alert* di Grafana dirancang untuk mendeteksi gangguan jaringan berdasarkan waktu respons *ping* yang melebihi ambang batas tertentu. Dalam hal ini, ambang batas waktu respons *ping* ditetapkan sebesar 30 *ms*. *Alert* akan aktif jika waktu respons *ping* konsisten melampaui nilai ini selama 5 menit berturut-turut. Durasi pengamatan selama 5 menit bertujuan untuk menghindari aktivasi *alert* akibat fluktuasi sesaat yang bersifat sementara. Dengan pengaturan ini, sistem monitoring dapat memberikan notifikasi yang relevan dan akurat, memprioritaskan deteksi gangguan jaringan yang signifikan tanpa memicu *false positive* akibat perubahan kecil yang normal pada jaringan.

#### 4.13.3 Implementasi *Alert* di Grafana

Langkah-langkah untuk mengatur *alert* di Grafana dimulai dengan membuka dasbor yang memonitor metrik waktu respons *ping*. Pada dasbor tersebut, pilih panel yang

menampilkan data *ping*, lalu klik Edit untuk membuka pengaturan panel. Setelah itu, tambahkan aturan *alert* pada tab *Alert* dengan membuat *query* yang menghitung rata-rata waktu respons *ping* menggunakan fungsi seperti `avg_over_time(ping_response[5m])`. Kemudian, atur kondisi agar *alert* aktif jika rata-rata waktu respons melebihi 30 *ms*. Untuk memastikan *alert* hanya aktif jika masalah berlangsung secara konsisten, pengaturan durasi pengamatan atau evaluasi *alert* diatur selama 5 menit berturut-turut. Selanjutnya, tambahkan *channel* notifikasi Telegram untuk memastikan tim jaringan menerima peringatan secara langsung ketika kondisi *alert* terpenuhi. Dengan konfigurasi ini, Grafana dapat memberikan notifikasi yang tepat waktu dan relevan untuk mendeteksi gangguan jaringan yang signifikan.

#### 4.13.4 Pemilihan Ambang Batas 30 *ms*

Ambang batas waktu respons *ping* sebesar 30 *ms* dipilih berdasarkan hasil analisis data variabilitas *ping* sebelumnya. Pada kondisi normal, rata-rata waktu respons *ping* berada di kisaran 25–26 *ms*, seperti yang tercatat pada tanggal 3 September 2024 dan 15 November 2024. Nilai ini menunjukkan performa jaringan yang stabil dengan fluktuasi minimal. Pada saat terjadi beban jaringan tinggi, seperti pada tanggal 13 November 2024, waktu respons tercatat mencapai hingga 40 *ms*, namun kondisi ini dianggap sebagai *outlier* yang tidak mencerminkan kondisi jaringan secara keseluruhan. Oleh karena itu, ambang batas 30 *ms* ditetapkan untuk memberikan sensitivitas yang cukup dalam mendeteksi gangguan signifikan, tanpa terlalu sering memicu *alert* akibat fluktuasi minor atau aktivitas normal. Dengan ambang batas ini, sistem monitoring dapat secara efektif membedakan antara gangguan yang signifikan dan fluktuasi sesaat, sehingga memberikan peringatan yang relevan dan dapat diandalkan bagi tim jaringan.

#### 4.13.5 Interpretasi dan Penerapan

Pengaturan *alert* ini dirancang untuk memberikan keseimbangan antara sensitivitas dan stabilitas dalam pemantauan performa jaringan. Dengan ambang batas waktu respons 30 *ms*, dan durasi pengamatan 5 menit, sistem monitoring akan mampu mendeteksi perubahan performa jaringan yang signifikan tanpa memicu *alert* yang berlebihan akibat fluktuasi kecil atau sementara. Pendekatan ini memastikan bahwa *alert* hanya aktif untuk kondisi yang benar-benar memerlukan perhatian teknis, sehingga membantu tim jaringan memfokuskan respons mereka pada masalah yang relevan.

Pengaturan ini dapat diterapkan pada sistem monitoring menggunakan Grafana yang memantau metrik jaringan secara *real-time*. *Alert* dapat diintegrasikan dengan sistem notifikasi seperti email atau bot Telegram untuk memberikan peringatan langsung kepada tim teknis jika terjadi penurunan kualitas jaringan. Dengan integrasi semacam ini, respon terhadap potensi gangguan jaringan dapat dilakukan dengan cepat, sehingga meminimalkan dampak terhadap pengguna akhir.

Secara keseluruhan, pengaturan *alert* ini dirancang berdasarkan hasil analisis data variabilitas *ping* yang telah diuraikan sebelumnya, menjadikannya relevan dan adaptif terhadap karakteristik jaringan yang dipantau. Namun, pengaturan ini masih dapat disesuaikan lebih lanjut untuk memenuhi kebutuhan spesifik jaringan atau preferensi sensitivitas yang diinginkan oleh tim pengelola. Dengan pemantauan yang lebih akurat, stabilitas jaringan dapat terjaga, mendukung kelancaran aktivitas yang memerlukan koneksi andal.

#### 4.14 Kesimpulan Implementasi dan Hasil

Implementasi sistem pemantauan performa jaringan menggunakan Raspberry Pi 4 sebagai pengumpul data, Prometheus sebagai penyimpan data, dan Grafana sebagai alat visualisasi berhasil dilakukan sesuai dengan tujuan penelitian. Sistem ini mampu memberikan informasi *real-time* terkait performa jaringan, seperti waktu respons *ping*, *packet loss*, kecepatan unduh dan unggah, serta lalu lintas jaringan, yang divisualisasikan dalam bentuk dasbor yang informatif. Pengujian yang dilakukan di berbagai lokasi memberikan hasil yang mencerminkan kemampuan sistem dalam meningkatkan *robustness* dan *resilience* jaringan.

Hasil implementasi mengindikasikan bahwa:

a. Stabilitas Jaringan (*Robustness*)

Sistem monitoring ini memungkinkan tim NDC (Network Data Center) untuk mendeteksi tren memburuknya performa jaringan sebelum gangguan serius terjadi. Dari grafik visualisasi waktu respons *ping* atau *packet loss* di Grafana, gejala atau tanda-tanda awal dari penurunan performa dapat diidentifikasi, seperti tren peningkatan waktu respons atau fluktuasi yang semakin besar. Hal ini memungkinkan tim NDC untuk segera mengambil langkah-langkah perbaikan sebelum jaringan benar-benar bermasalah atau putus total. Sebelum sistem ini diterapkan, tim NDC hanya dapat mengetahui jika jaringan benar-benar terputus, tanpa memiliki kemampuan untuk mengidentifikasi tren memburuknya performa jaringan. Dengan fitur ini, *robustness* jaringan ditingkatkan karena masalah dapat ditangani

secara proaktif, menjaga stabilitas layanan jaringan meskipun menghadapi kondisi yang menantang.

b. Kemampuan Pemulihan (*Resilience*)

Sistem ini juga menunjukkan kemampuan *resilience* yang sangat baik. Dengan fitur notifikasi otomatis melalui bot Telegram, tim teknis dapat segera merespons gangguan seperti peningkatan waktu respons *ping* atau *packet loss* yang tinggi. *Resilience* ditingkatkan dengan kemampuan sistem untuk memberikan notifikasi dini dan data pendukung yang jelas, sehingga tindakan perbaikan dapat dilakukan sebelum gangguan menjadi signifikan. Selain itu, integrasi dengan dasbor Grafana mempermudah tim untuk mengevaluasi penyebab gangguan dan memulihkan performa jaringan dengan cepat, meminimalkan dampak gangguan terhadap pengguna.

c. Pengaruh Beban Jaringan

Aktivitas berat, seperti konferensi video atau acara besar, memicu lonjakan waktu respons dan *packet loss* yang signifikan. Sistem ini membantu mengidentifikasi bahwa lonjakan tersebut disebabkan oleh peningkatan jumlah perangkat yang terhubung dan aktivitas intensif, memberikan wawasan yang berguna bagi pengelolaan *bandwidth* dan pengaturan kapasitas jaringan.

d. Efektivitas Sistem Monitoring

Sistem berbasis Raspberry Pi 4, Prometheus, dan Grafana terbukti efektif dalam mendukung pengelolaan jaringan. Notifikasi *real-time* membantu tim teknis untuk merespons masalah secara cepat, sementara visualisasi data di dasbor Grafana memungkinkan analisis performa yang mendalam. Hal ini meningkatkan efisiensi operasional dan memastikan layanan jaringan tetap optimal.

e. Pengaturan Ambang Batas *Alert*

Penggunaan ambang batas waktu respons *ping* sebesar 30 ms terbukti efektif dalam membedakan fluktuasi minor dari gangguan signifikan. Ini membantu tim NDC memprioritaskan perhatian pada masalah yang benar-benar membutuhkan intervensi, sehingga sumber daya teknis dapat dikelola dengan lebih efisien.

Secara keseluruhan, implementasi sistem pemantauan ini telah memenuhi kebutuhan penelitian. Sistem ini tidak hanya memberikan solusi untuk memantau dan menganalisis performa jaringan secara *real-time*, tetapi juga meningkatkan *robustness* dengan mendeteksi tren memburuknya performa jaringan, serta *resilience* dengan memberikan notifikasi dini untuk gangguan potensial. Dengan hasil ini, sistem ini memiliki potensi untuk dikembangkan

lebih lanjut, termasuk optimalisasi jaringan dan penerapan di skala yang lebih luas. Sistem ini menjadi alat penting dalam memastikan kualitas layanan jaringan yang andal, mendukung aktivitas akademik dan operasional yang membutuhkan koneksi stabil.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melalui serangkaian penelitian, analisis, dan implementasi, beberapa poin penting yang dapat diambil sebagai kesimpulan dari penelitian ini adalah sebagai berikut:

a. Pengaruh Teknologi *Edge Sensors* terhadap Jaringan *Enterprise*

Penerapan teknologi *edge sensors* dalam sistem jaringan *enterprise* terbukti memberikan dampak positif terhadap peningkatan *robustness* dan *resilience* jaringan. Penggunaan *edge sensors* memungkinkan deteksi dini masalah yang mempengaruhi kinerja jaringan, baik dari sisi ketahanan (*robustness*) terhadap gangguan, maupun kemampuan jaringan untuk pulih dengan cepat (*resilience*).

b. *Robustness* Jaringan di Lingkungan Kampus

*Robustness* jaringan nirkabel di lingkungan kampus dapat diketahui melalui pemantauan metrik performa seperti *ping*, *latency*, dan *packet loss* secara *real-time*. Dalam penelitian ini, sistem pemantauan berbasis *edge sensors* terbukti efektif dalam mendeteksi adanya anomali atau gangguan performa sebelum gangguan tersebut memengaruhi kualitas layanan jaringan secara signifikan. Dengan menggunakan metrik-metrik ini, administrator dapat mengetahui kapan dan di mana gangguan terjadi, serta menganalisis faktor-faktor penyebabnya. Dengan pemantauan yang terus-menerus, *robustness* jaringan dapat dijaga dengan baik, sehingga jaringan tetap stabil meskipun terjadi fluktuasi atau variasi kondisi yang tidak terduga.

c. *Resilience* Jaringan dan Pengaruhnya terhadap Pemulihan Gangguan

*Resilience* jaringan di lingkungan kampus dapat ditingkatkan melalui penerapan sistem notifikasi otomatis yang dikendalikan oleh *edge sensors*. Dalam penelitian ini, *edge sensors* membantu mendeteksi masalah performa lebih awal dan memberikan notifikasi kepada administrator jaringan segera setelah metrik kinerja melebihi ambang batas yang telah ditentukan. Dengan adanya peringatan dini ini, respons terhadap gangguan dapat dilakukan dengan lebih cepat, memungkinkan jaringan untuk pulih lebih cepat setelah terjadi gangguan. Ini membuktikan bahwa *edge sensors* berperan dalam mempercepat pemulihan jaringan, meminimalkan dampak gangguan, dan memastikan kelangsungan layanan jaringan dalam waktu yang singkat.

d. Efisiensi Pemantauan Jaringan melalui Raspberry Pi 4

Raspberry Pi yang digunakan sebagai *node* pemantauan dalam jaringan terbukti menjadi solusi yang efisien dan hemat biaya dalam mengumpulkan data metrik seperti *ping*, *latency*, dan *packet loss*. Pemanfaatan Raspberry Pi sebagai *edge sensor* memungkinkan pemantauan jaringan secara real-time dengan perangkat yang berdaya rendah, tanpa mengorbankan akurasi pemantauan dan performa jaringan.

e. Visualisasi Data dengan Prometheus dan Grafana

Kombinasi antara Prometheus sebagai sistem monitoring dan Grafana sebagai platform visualisasi data terbukti efektif dalam menyajikan informasi performa jaringan. Dasbor yang interaktif memungkinkan pemantauan jaringan secara mudah, sehingga gangguan dapat diidentifikasi lebih cepat. Visualisasi data yang akurat memudahkan administrator jaringan dalam mengambil keputusan berbasis data untuk menjaga *robustness* dan *resilience* jaringan..

f. Manajemen *Alert* yang Responsif

Pengaturan sistem *alert* berbasis metrik performa jaringan, seperti ambang batas *ping* dan *packet loss*, memungkinkan deteksi dini terhadap potensi gangguan jaringan. Sistem ini terbukti mampu memberikan notifikasi secara *real-time* kepada penanggung jawab jaringan, sehingga meminimalkan waktu henti dan meningkatkan kualitas layanan jaringan.

g. Stabilitas Jaringan di Lingkungan Kampus

Pengujian pada jaringan di lingkungan kampus menunjukkan bahwa sistem ini dapat meningkatkan stabilitas jaringan, terutama di area yang memiliki jumlah pengguna yang tinggi. Dengan adanya pemantauan yang berkelanjutan, sistem ini juga mampu mengurangi *downtime* secara signifikan, serta memastikan jaringan yang *robust* dan *resilient* dalam menghadapi tantangan operasional sehari-hari.

## 5.2 Saran

Berdasarkan hasil penelitian ini, ada beberapa saran untuk pengembangan dan penerapan lebih lanjut:

a. Peningkatan Skala dan Cakupan Implementasi

Pengembangan sistem ini perlu diperluas ke seluruh area kampus atau bahkan ke jaringan dengan skala yang lebih besar, seperti jaringan perusahaan atau institusi lainnya. Hal ini akan memberikan wawasan lebih luas mengenai efisiensi dan efektivitas sistem di berbagai lingkungan.

b. Integrasi Teknologi Otomasi

Untuk meningkatkan efisiensi, integrasi teknologi otomasi jaringan dapat dipertimbangkan. Misalnya, dengan menggunakan skrip otomatis untuk menjalankan tindakan korektif ketika mendeteksi gangguan, sehingga tidak memerlukan intervensi manual setiap saat.

c. Pemantauan Kinerja di Berbagai Kondisi Jaringan

Untuk memastikan keandalan sistem di berbagai skenario, perlu dilakukan pengujian di kondisi jaringan yang lebih kompleks, seperti pada waktu puncak penggunaan atau pada jaringan yang lebih rentan terhadap gangguan. Hal ini akan memastikan bahwa sistem monitoring tetap tangguh dalam situasi yang dinamis.

d. Pengembangan Mekanisme Pemberitahuan yang Lebih Adaptif

Sistem pemberitahuan dapat dikembangkan lebih lanjut dengan memperkenalkan mekanisme pemberitahuan yang adaptif, seperti notifikasi berbasis prioritas atau tingkat keparahan masalah. Ini akan membantu pengelola jaringan dalam memfokuskan upaya pada masalah yang paling kritis terlebih dahulu.

e. Dengan demikian, sistem yang telah dikembangkan dalam penelitian ini memberikan dasar yang kuat untuk pemantauan jaringan yang efisien dan dapat terus ditingkatkan melalui berbagai pengembangan di masa mendatang.

## DAFTAR PUSTAKA

- Alpeni, R., Hadi, I., & Soim, S. (2019). Aplikasi Monitoring Wifi.Id PT Telekomunikasi Indonesia Berbasis Android. *Jurnal Riset Sistem Informasi Dan Teknik Informatika (JURASIK)*, 4, 171–180. <http://tunasbangsa.ac.id/ejurnal/index.php/jurasik>
- Brendgen, D. (2024). *Ping Exporter*. [https://github.com/czerwonk/ping\\_exporter](https://github.com/czerwonk/ping_exporter)
- Carvalho, M. N. de. (2023, June 29). *Speedtest Exporter*. <https://github.com/MiguelNdeCarvalho/speedtest-exporter>
- DATASHEET Raspberry Pi 4 Model B*. (2024). <https://www.raspberrypi.org>
- Fu, X., Yao, H., & Yang, Y. (2021). Modeling and optimizing the cascading robustness of multisink wireless sensor networks. *IEEE Transactions on Reliability*, 70(1), 121–133. <https://doi.org/10.1109/TR.2020.3024797>
- Hannats, M., Ichsan, H., & Korespondensi, P. (2021). ANALISIS KINERJA JARINGAN SENSOR NIRKABEL UNTUK EDGE COMPUTING MENGGUNAKAN LORA SX1278. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, Vol 8(No 5). <https://doi.org/10.25626/jtiik.202183631>
- Jones, H. W. (2021). *Going Beyond Reliability to Robustness and Resilience in Space Life Support Systems Nomenclature CCF = Common Cause Failure ECLSS = Environmental Control and Life Support System ESM = Equivalent System Mass ISS = International Space Station*.
- Kakadia, D., & Ramirez-Marquez, D. J. E. (2020). Quantitative approaches for optimization of user experience based on network resilience for wireless service provider networks. *Reliability Engineering and System Safety*, 193. <https://doi.org/10.1016/j.ress.2019.106606>
- Kelly, W. (2022, December). *Wireless Pccess Point*. TechTarget. <https://www.techtarget.com/searchmobilecomputing/definition/access-point>
- Khairullah, S. M. (2024). *IMPLEMENTASI PERANCANGAN SENTRALISASI SSID MENGGUNAKAN CONTROLLER ARUBA INSTANT ON PADA JARINGAN NIRKABEL PT. JARVIS INTEGRASI SOLUSI* [STT Terpadu Nurul Fikri]. <https://repository.nurulfikri.ac.id/id/eprint/621/>
- Kong, Z., & Yeh, E. M. (2009). *Wireless Network Resilience to Degree-Dependent and Cascading Node Failures*.

- Li, J., Wang, Y., Zhong, J., Sun, Y., Guo, Z., Chen, Z., & Fu, C. (2022). Network resilience assessment and reinforcement strategy against cascading failure. *Chaos, Solitons and Fractals*, 160. <https://doi.org/10.1016/j.chaos.2022.112271>
- Mottahedi, A., Sereshki, F., Ataei, M., Nouri Qarahasanlou, A., & Barabadi, A. (2021). Resilience analysis: A formulation to model risk factors on complex system resilience. *International Journal of System Assurance Engineering and Management*, 12(5), 871–883. <https://doi.org/10.1007/s13198-021-01131-w>
- Pajic Miroslav, S. S. L. N. J. J. P. G. and M. R. (2010). *The Wireless Control Network: Synthesis and Robustness*. IEEE.
- Potdar, A. M., Narayan, D. G., Kengond, S., & Mulla, M. M. (2020). Performance Evaluation of Docker Container and Virtual Machine. *Procedia Computer Science*, 171, 1419–1428. <https://doi.org/10.1016/j.procs.2020.04.152>
- Sukhija, N., & Bautista, E. (2019). Towards a framework for monitoring and analyzing high performance computing environments using kubernetes and prometheus. *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, 257–262. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00087>
- Ushakova, I., Plokha, O., & Skorin, Y. (2022). Approaches to web application performance testing and real-time visualization of results. *Bulletin of Kharkov National Automobile and Highway University*, 96, 71. <https://doi.org/10.30977/bul.2219-5548.2022.96.0.71>
- Wang, C. Y., Huang, H. Y., & Hwang, R. H. (2011). Mobility management in ubiquitous environments. *Personal and Ubiquitous Computing*, 15(3), 235–251. <https://doi.org/10.1007/s00779-010-0328-2>
- Williams, M. (2024, July 11). *Tailscale VPN review*. TechRadar.
- Xu, F., Tan, C. C., Li, Q., Yan, G., & Wu, J. (2010). *Designing a Practical Access Point Association Protocol*.
- Ziemke, J. (2020). *Node Exporter*. [https://github.com/prometheus/node\\_exporter/](https://github.com/prometheus/node_exporter/)
- Zou, Y., Yu, D., Yu, J., Zhang, Y., Dressler, F., & Cheng, X. (2021). Distributed Byzantine-Resilient Multiple-Message Dissemination in Wireless Networks. *IEEE/ACM Transactions on Networking*, 29(4), 1662–1675. <https://doi.org/10.1109/TNET.2021.3069324>

## **LAMPIRAN**