

BAB 3

METODE PENELITIAN

3.1 PERANGKAT YANG DIGUNAKAN

3.1.1 PERANGKAT KERAS (*HARDWARE*)

Pada penelitian ini menggunakan perangkat keras (*hardware*) berupa 1 buah laptop yang digunakan untuk melakukan simulasi dan pengolahan data, spesifikasi dari *hardware* tersebut terdapat pada tabel 3.1.

Tabel 3.1 Spesifikasi *Hardware*

OS	Ubuntu 22.10
<i>Processor</i>	AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx 2.60 GHz
RAM	8 GB

3.1.2 PERANGKAT LUNAK (*SOFTWARE*)

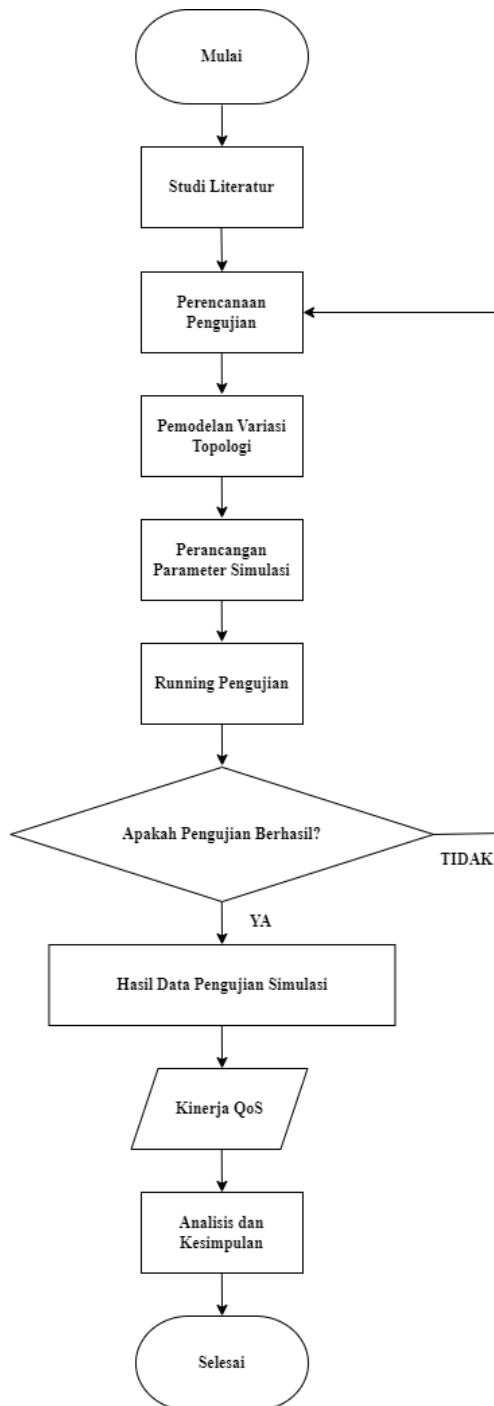
Terdapat perangkat lunak (*software*) dan juga *software tools* yang digunakan untuk membantu melakukan simulasi dalam penelitian ini, perangkat lunak (*software*) dan *tools* tersebut terdapat pada tabel 3.2.

Tabel 3.2 *Software dan Tools*

No	<i>Software</i>	Versi	Fungsi
1.	NSG	2.1	<i>Tools</i> yang menghasilkan TCL <i>script</i> untuk menjalankan NS2
2.	<i>Network Simulator</i>	2.35	<i>Software Simulator</i>

3.2 ALUR PENELITIAN

Penelitian ini melalui beberapa tahap yang harus dilakukan untuk menyelesaikannya. Sehingga, diagram alur penelitian yang dilakukan ditunjukkan pada gambar 3.1.



Gambar 3.1 Alur Penelitian

Gambar 3.1 menunjukkan alur penelitian atau *flowchart* yang dilakukan oleh penulis. Dimulai dengan studi literatur yang dilakukan untuk mencari, mengumpulkan, dan memahami teori yang berkaitan dengan penelitian yang dibuat. Sehingga, penulis dapat membandingkan beberapa jurnal yang berkaitan dan nantinya akan digunakan sebagai pembanding untuk memahami konsep dasar dari topik penelitian. Dalam hal ini, penulis mengambil sumber dari beberapa jurnal, buku, dan *website* sebagai dasar untuk mengidentifikasi serta mengevaluasi pengetahuan yang sudah ada dalam bidang yang bersangkutan. Selain itu, tahap studi literatur ini membantu peneliti untuk memahami konteks dan latar belakang topik penelitian sehingga lebih paham dengan perkembangan sebelumnya dan isu – isu yang relevan.

Kemudian tahap perencanaan pengujian dilakukan untuk memfokuskan pada desain dan implementasi pengujian pada penelitian yang akan dilakukan. Penulis dapat mengidentifikasi mengenai penggunaan *routing protocol* DSDV pada topologi jaringan yang diterapkan pada infrastruktur. Selain itu, pada tahap ini penulis juga menentukan metodologi pengujian yang sesuai, sehingga membantu peneliti untuk mencapai tujuan penelitian dengan cara terstruktur dan terorganisir.

Selanjutnya pada tahap pemodelan topologi, penulis melakukan proses pembuatan representasi visual dari jaringan yang menunjukkan bagaimana komponen – komponen terhubung dan berpengaruh satu sama lain. Dalam pemodelan topologi, komponen – komponen sistem jaringan dapat diwakili sebagai simpul atau node, sedangkan hubungan antar simpul dapat diwakili sebagai garis atau tautan. Pemodelan topologi dapat menyediakan informasi penting untuk memahami dan mengatasi masalah jaringan, seperti menentukan lokasi kegagalan ketika dilakukan simulasi pengambilan hasil data. Pada pemodelan topologi ini penulis, membuat pemodelan dengan model topologi *star*, *tree*, dan *mesh*.

Untuk tahap perancangan parameter simulasi ini dilakukan untuk memfokuskan tujuan simulasi sesuai dengan kebutuhan dan mencapai hasil yang diinginkan yaitu menganalisis kinerja topologi jaringan *wireless* dengan beberapa variasi yang terdapat pada penelitian ini. Dengan tahap ini, penulis dapat memaksimalkan manfaat dari hasil simulasi, memberikan kontribusi yang berarti

dalam pengembangan dan penelitian, serta meningkatkan pemahaman mengenai topik penelitian ini karena dengan mendokumentasikan parameter simulasi secara rinci, maka mudah untuk mereplikasi atau mengembangkan simulasi yang dibuat. Hasil data dari simulasi yang dilakukan ini penting dalam validasi dan verifikasi temuan serta memastikan keandalan penelitian.

Tahap selanjutnya adalah *running test* yang digunakan untuk pengujian dan mengevaluasi performa dan kinerja perancangan tahap sebelumnya. Tahap ini digunakan untuk menentukan apakah pengujian simulasi berhasil dan sesuai yang diharapkan. Dengan menjalankan *script* yang telah dibuat, maka akan menjalankan simulasi sesuai dengan konfigurasi yang telah ditentukan. Selama simulasi berjalan dapat memantau proses dengan melihat *output* yang dihasilkan. Apabila pengujian tersebut tidak berhasil maka akan kembali pada tahap perancangan pengujian untuk menganalisis kesalahan sehingga *output* tidak sesuai dengan apa yang diinginkan, apabila pengujian berhasil maka akan lanjut pada tahap selanjutnya yaitu menghitung nilai *delay*, *throughput*, dan *packet loss*.

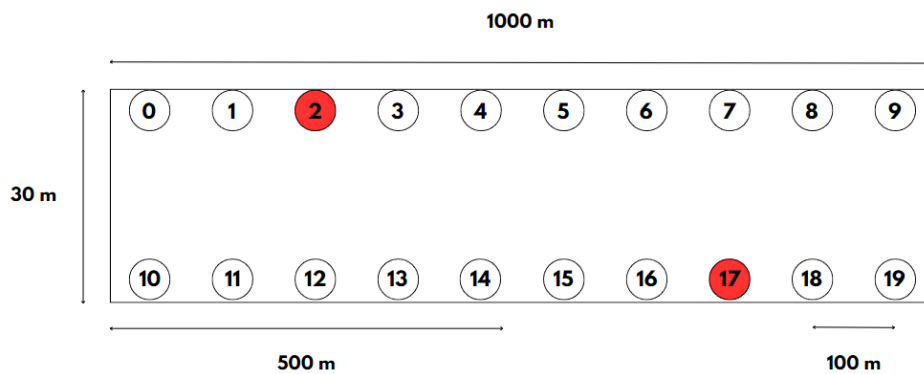
Output yang dihasilkan dari pengujian simulasi akan menghasilkan nilai yang untuk perhitungan *delay*, *throughput*, dan *packet loss*. Parameter *delay* merupakan waktu yang diperlukan oleh paket data untuk mencapai tujuan dari pengirim ke penerimanya. Apabila waktu yang dibutuhkan paket data untuk mencapai tujuan dari pengirim ke penerima sebanyak $< 150s$ maka termasuk kedalam kategori *perfect* atau bisa dikatakan semakin kecil nilainya maka akan semakin baik. *Throughput* mengacu pada jumlah data yang dapat ditransmisikan dalam satu periode waktu, dimana semakin besar nilainya maka akan semakin baik dengan kategori *excellent*. Lalu untuk nilai dari *packet loss* akan semakin baik apabila nilainya semakin kecil yang dimana *packet loss* digunakan untuk mengukur seberapa banyak paket data yang hilang selama pengiriman, nilai ini diukur sebagai presentasi hilangnya paket dalam lalu lintas jaringan.

Setelah perhitungan selesai dan hasil data yang sesuai, maka tahap terakhir ini melakukan analisis dan mengambil kesimpulan yang didapatkan dari penelitian ini berdasarkan hasil data yang ada. Pada tahap ini, peneliti menganalisis dengan metode komparatif dimana membandingkan kinerja beberapa topologi jaringan dengan nilai *Quality of Service* (QoS) yang berbeda serta memberikan visualisasi berupa grafik dan interpretasi hasil analisis, sehingga data akan lebih mudah dipahami dan jelas bagaimana data tersebut relevan dengan konteks atau tujuan

penelitian dan pengujian yang akan dilakukan.

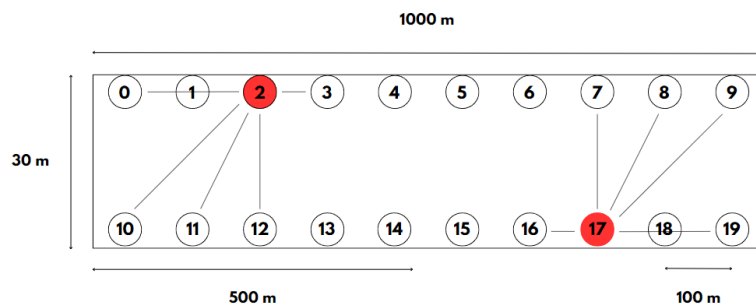
3.3 PERANCANGAN TOPOLOGI JARINGAN

Pada penelitian ini, pemodelan topologi yang dibuat adalah topologi *star*, *tree*, dan *mesh*. Topologi tersebut diterapkan pada sebuah infrastruktur yaitu jembatan yang memiliki panjang 1 km atau 1000 m dan lebar 30 m. Skema penempatan node dapat dilihat pada gambar 3.2.

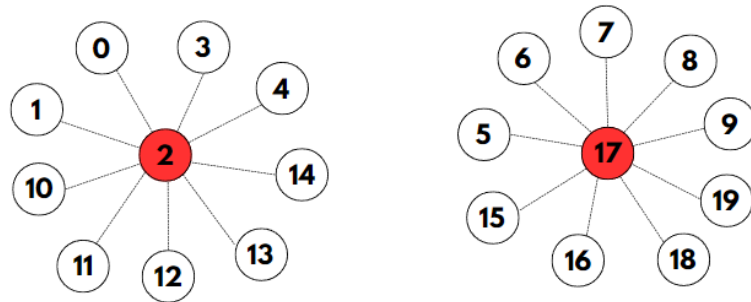


Gambar 3.2 Skema Penempatan Node Dalam Infrastruktur Jembatan

Dengan panjang dan lebar yang telah didefinisikan, maka pada infrastruktur tersebut terdapat 20 node yang digunakan, dengan 18 node sensor yang berfungsi untuk membaca data pada area pemantauan (node sensor) dan 2 node sink yang diletakkan pada node 2 dan node 17 sebagai node yang mengumpulkan data penginderaan dari node sensor. Sehingga, untuk penempatan node jarak antar setiap nodenya adalah 100 m.

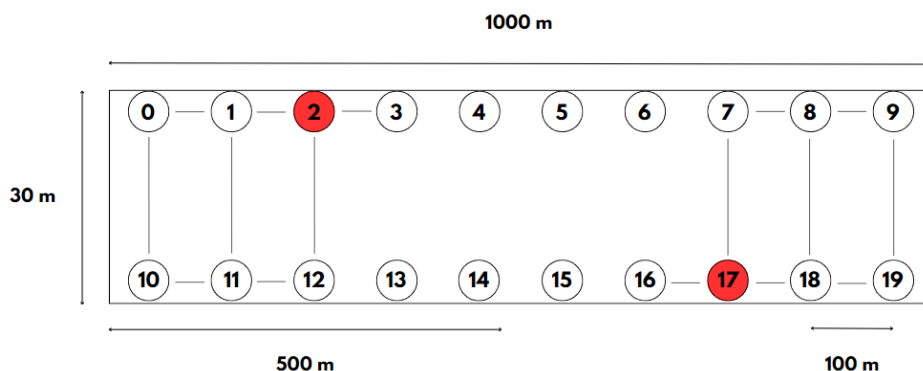


Gambar 3.3 Skema Topologi Star Dalam Infrastruktur Jembatan

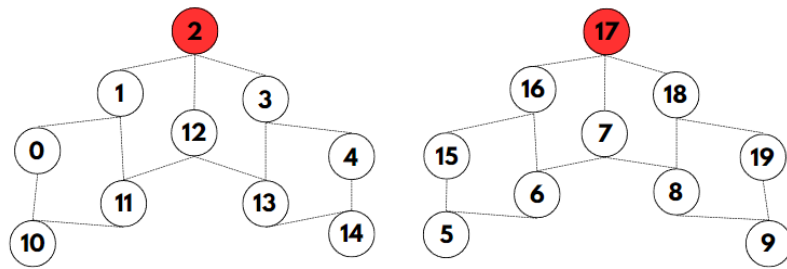


Gambar 3.4 Skema Topologi *Star*

Untuk pemodelan topologi *star* untuk skema yang diimplementasikan dalam infrastruktur dan gambar 3.3 untuk skema topologi *star* yang digambarkan tanpa diimplementasikan dalam infrastruktur, setiap node sensor dapat terhubung dengan node sink dimana node 2 sebagai node sink dapat terhubung dengan node sensor 0, node sensor 1, node sensor 3, node sensor 4, node sensor 10, node sensor 11, node sensor 12, node sensor 13, dan node sensor 14. Sedangkan node sink 17, dapat terhubung langsung dengan node sensor 5, node sensor 6, node sensor 7, node sensor 8, node sensor 9, node sensor 15, node sensor 16, node sensor 18, dan node sensor 19. Topologi *star* dapat dilihat pada gambar 3.4. Pembagian node sensor ke node sink tersebut bertujuan agar jarak atau jalur komunikasi antar node tidak terlalu jauh, sehingga proses transmisi data dapat lebih baik. Karena semakin jauh jarak node sensor dan node sinknya, maka paket yang hilang akan semakin besar.

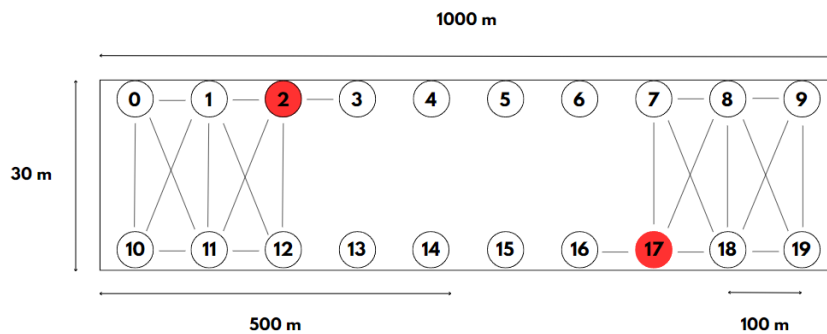


Gambar 3.5 Skema Topologi *Tree* Dalam Infrastruktur Jembatan

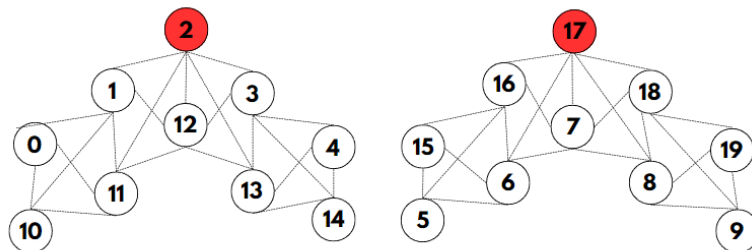


Gambar 3.6 Skema Topologi *Tree*

Selanjutnya untuk skema topologi *tree* pada penelitian ini dapat dilihat pada gambar 3.5 untuk skema topologi dalam infrastruktur dan gambar 3.6 skema topologi yang tidak diimplementasikan pada infrastruktur. Setiap node pada topologi tersebut dibuat untuk dapat berkomunikasi dengan node di sebelahnya dan node bersebelahan yang berada disebaliknya. Sebagai contoh pada node sensor 0 yang dapat berkomunikasi dengan node sensor 1 dan juga node sensor 10; node sensor 8 dapat berkomunikasi dengan node sensor 7, node sensor 9, dan node sensor 18 yang berada disebaliknya.



Gambar 3.7 Skema Topologi *Mesh* Dalam Infrastruktur Jembatan



Gambar 3.8 Skema Topologi *Mesh*

Untuk topologi *mesh*, terdapat pada gambar 3.7 untuk skema topologi yang diimplementasikan pada infrastruktur dan gambar 3.8 untuk skema topologi yang tidak diimplementasikan pada infrastruktur. Setiap node dalam topologi ini dapat berkomunikasi secara vertikal, horizontal, dan diagonal dengan node yang berdekatan. Sebagai contoh untuk node sensor 0 dapat berkomunikasi dengan node sensor 1, node sensor 10, dan node sensor 11; dan seterusnya.

3.4 SKENARIO PENGUJIAN

3.4.1 PERANCANGAN SKENARIO

Pada perancangan skenario dalam penelitian ini menggunakan variasi tiga pemodelan topologi yaitu topologi *star*, *tree*, dan *mesh* yang memiliki 20 node dan menggunakan *routing protocol* DSDV untuk setiap topologinya. Tabel 3.3 merupakan skenario yang dilakukan dalam pengujian.

Tabel 3.3 Skenario Penelitian

<i>Routing Protocol</i>	Topologi	<i>Data Rate</i>	Pengujian
DSDV	<i>Star</i>	250 kbps	20 kali
	<i>Tree</i>		
	<i>Mesh</i>		

Perancangan pada penelitian ini akan dilakukan sebanyak 20 kali percobaan dengan menggunakan *routing protocol* DSDV pada setiap percobaannya. Tiap skenario memiliki *data rate* yaitu sebesar 250 kbps. *Data rate* mengacu pada kecepatan maksimum dimana data dapat ditransmisikan dalam jaringan yang mengacu pada pengambilan teknologi komunikasi menggunakan Zigbee. Teknologi Zigbee mendukung beberapa tingkat *data rate* yang berbeda, yang umumnya disesuaikan dengan kebutuhan jaringan dan aplikasi tertentu. *Data rate* yang lebih tinggi memungkinkan untuk mentransfer data lebih cepat, sementara

data rate yang lebih rendah mungkin lebih cocok untuk aplikasi yang memerlukan hemat energi atau operasi dalam jarak yang lebih jauh. Penggunaan *routing protocol* DSDV yang mengandalkan tabel *routing* yang setiap node sensor dalam jaringan memiliki tabel *routing* yang diperbarui secara berkala, sehingga dapat menentukan jalur terbaik pada jaringan.

Skenario pertama dengan model topologi *star* yang memiliki simpul pusat (node sink) yang terhubung ke semua node sensornya dan semua node tersebut berkomunikasi dengan node sink secara langsung. Performa jaringan mungkin lebih baik pada *data rate* yang lebih tinggi jika simpul pusat memiliki kapasitas dan sumber daya yang cukup. Hal ini berkaitan dengan kapasitas dan sumber daya yang tersedia di simpul pusat serta faktor – faktor lain yang mempengaruhi efisiensi jaringan salah satunya peningkatan kualitas layanan *Quality of Service* (Qos). Dengan memiliki kapasitas yang lebih besar di simpul pusat, jaringan memiliki kemampuan untuk menangani lebih banyak data secara bersamaan. Ketika *data rate* meningkat, jaringan perlu mengakomodasikan peningkatan *traffic* yang lebih besar. Node sink dengan kapasitas yang cukup besar dapat memproses dan mengarahkan data dengan lebih efisien.

Kemudian skenario kedua menguji topologi *tree* yang strukturnya hierarkis di mana simpul – simpul sensor terhubung ke simpul lain yang bertindak sebagai *parent* node, dan akhirnya mencapai simpul pusat. Pada topologi *tree*, performa akan sangat bergantung pada kedalaman dan struktur tree Pada *data rate* yang lebih tinggi, potensi konvergensi lama dari simpul *leaf* ke simpul pusat mungkin mempengaruhi latensi atau mempengaruhi lama waktu pengumpulan data oleh node sensor ke node sink. Pemilihan struktur *tree* yang efisien dapat mempengaruhi performa pada jaringan. Salah satu karakteristik dari struktur tree yang efisien yaitu memilih jalur yang optimal karena setiap simpul dalam struktur *tree* memiliki jalur yang digunakan untuk mengirimkan dan menerima data. Jalur ini dirancang untuk, mengurangi latensi, memaksimalkan kecepatan transfer data, dan menghindari lompatan yang tidak perlu dimana data harus melewati simpul – simpul atau perangkat jaringan tambahan yang sebenarnya tidak diperlukan dalam perjalanan menuju tujuan akhir.

Skenario ketiga menggunakan model topologi *mesh* dimana semua simpul

sensor terhubung satu sama lain, membentuk jaringan yang padat. Topologi *mesh* umumnya memiliki redundansi atau konsep di mana elemen atau komponen tambahan diberikan ke dalam sistem atau struktur dengan tujuan untuk mengatasi potensi kegagalan, meningkatkan keandalan, dan menjaga kontinuitas operasi; memiliki nilai yang baik, sehingga kerusakan satu atau beberapa simpul tidak akan langsung mengganggu seluruh jaringan. Namun, semakin besar jumlah simpul, maka semakin kompleks juga manajemen *routing*-nya. Pada *data rate* yang lebih tinggi, masalah latensi dan *overhead routing protocol* mungkin timbul karena kompleksitas jaringan yang lebih besar karena pengaturan yang salah atau kurang tepat akan berdampak negatif pada kinerja jaringan.

3.4.2 PERANCANGAN PARAMETER SIMULASI

Penelitian ini terdapat beberapa parameter yang digunakan untuk melakukan simulasi pengujian. Parameter simulasi tersebut ditunjukkan pada tabel 3.4 yang juga mencantumkan parameter uji *delay*, *throughput*, dan *packet loss*.

Tabel 3.4 Parameter Simulasi

Topologi	<i>Data Rate</i>	Parameter		
Star	250 kbps	<i>Delay</i>	<i>Throughput</i>	<i>Packet Loss</i>
Tree				
Mesh				

Pada penelitian ini, perancangan simulasi dilakukan untuk mengatur kondisi dan karakteristik dari simulasi. Tujuan dari perancangan simulasi yaitu memodelkan dan memahami perilaku jaringan atau sistem yang akan dievaluasi, tanpa harus melibatkan pengujian langsung di lingkungan fisik yang sebenarnya. Tiap topologi memiliki TCP (*Transmission Control Protocol*) biasanya aplikasi yang berjalan pada node sensor dan node sink menggunakan TCP untuk mengirim dan menerima data. TCP dapat digunakan untuk mengatur koneksi antar node sensor dan node sink, serta memastikan pengiriman data yang andal dan teratur. Node sensor akan

bertindak sebagai klien yang mengirimkan data ke node sink sebagai server. Dalam aplikasi ini, implementasi TCP berada di lapisan aplikasi pada kedua node (sensor dan sink). Ketika node sensor ingin mengirimkan data, aplikasi pada node sensor akan berkomunikasi dengan aplikasi pada node sink melalui protokol TCP.

Penggunaan *data rate* atau tingkat kecepatan transfer ini merupakan parameter yang menentukan seberapa cepat data dapat dikirim melalui jaringan. *Data rate* diukur dalam kilobit per detik (kbps) dan menunjukkan jumlah bit data yang dapat ditransfer dalam satu detik. Semakin tinggi *data rate*, semakin cepat data dapat dikirim, dan sebaliknya. Dalam simulasi ini menggunakan *data rate* 250 kbps yang akan diterapkan pada tiap topologi jaringan yang akan diuji. Pengambilan variasi *data rate* ini berdasarkan salah satu teknologi komunikasi yaitu Zigbee yang dimana dirancang khusus untuk aplikasi jaringan sensor nirkabel dan memiliki karakteristik yang memungkinkan jaringan sensor nirkabel dengan konsumsi daya rendah, jangkauan jaringan yang luas, dan kapasitas jaringan yang tinggi. Melakukan variasi *data rate* dalam pengujian kinerja topologi jaringan *wireless* akan membantu melakukan evaluasi pengaruh kecepatan transmisi data pada WSN yang diimplementasikan dengan protokol DSDV dan teknologi Zigbee tersebut. Perubahan *data rate* dapat mempengaruhi *throughput*, *delay*, dan *packet loss* di jaringan, sehingga penulis dapat memahami performa dalam situasi yang berbeda dan menentukan *data rate* yang sesuai untuk diterapkan pada infrastruktur jembatan yang digunakan untuk melakukan SHM.

Parameter pengujian kinerja topologi menggunakan analisis Qos bertujuan untuk mengevaluasi performa jaringan dan memastikan bahwa jaringan berfungsi secara efisien dan andal. Parameter *delay* mengukur waktu yang dibutuhkan untuk data dikirimkan dari sumber ke tujuan melalui jaringan. Pada jaringan *real-time* seperti VoIP atau *video streaming*, *delay* yang tinggi dapat menyebabkan kualitas layanan yang buruk dan interaksi yang tidak lancar. Oleh karena itu, mengukur dan mengamati *delay* dalam pengujian kinerja penting untuk memastikan bahwa jaringan dapat mengirimkan data dengan waktu respons yang sesuai. *Throughput* digunakan untuk mengukur jumlah data yang berhasil dikirim atau diterima dalam satuan waktu tertentu. Pengujian *throughput* pada jaringan *wireless* dengan modul Zigbee berguna untuk mengetahui berapa banyak data yang dapat diakomodasi oleh

jaringan dalam kondisi tertentu. Ini membantu dalam menentukan sejauh mana jaringan mampu menangani lalu lintas data yang dihasilkan oleh aplikasi atau pengguna. *Packet loss* adalah persentase paket data yang hilang selama proses pengiriman melalui jaringan. Hal ini dapat terjadi karena kegagalan transmisi, kepadatan lalu lintas, atau masalah jaringan lainnya. *Packet loss* dapat menyebabkan degradasi atau penurunan performa aplikasi, terutama pada aplikasi *real-time* yang sensitif terhadap kehilangan data. Pengujian *packet loss* membantu mengidentifikasi masalah dan mengevaluasi seberapa handal jaringan dalam mengirimkan data.

Dengan mengambil parameter QoS seperti *delay*, *throughput*, dan *packet loss* pada pengujian kinerja, dapat menilai keandalan dan efisiensi jaringan *wireless* dengan protokol DSDV dan modul Zigbee pada nodenya. Data dan hasil dari pengujian ini dapat digunakan untuk mengoptimalkan jaringan, mengidentifikasi titik lemah, dan meningkatkan performa keseluruhan jaringan guna mencapai kualitas layanan yang lebih baik bagi pengguna akhir.