



TESIS - TE142599

**DESAIN DAN ANALISA MANAJEMEN KONSUMSI
DAYA PADA WSN UNTUK SISTEM MONITORING
KESEHATAN STRUKTUR (SMKS) JEMBATAN**

Faridatun Nadziroh
2213203018

DOSEN PEMBIMBING
Eko Setijadi, ST, MT, Ph.D
Dr. Ir. Wirawan, DEA

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015



TESIS - TE142599

**DESIGN AND ANALYSIS OF POWER
CONSUMPTION MANAGEMENT WSN FOR
BRIDGE STRUCTURAL HEALTH
MONITORING SYSTEM (SHMS)**

Faridatun Nadziroh
2213203018

Supervisor
Eko Setijadi, ST, MT, Ph.D
Dr. Ir. Wirawan, DEA

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2015

Tesis telah disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (MT)

Di
Institut Teknologi Sepuluh Nopember

oleh :
Faridatun Nadziroh
NRP. 2213203018

Tanggal Ujian : 17 Juni 2015

Periode Wisuda : September 2015

Disetujui oleh:

1. **Eko Setidjadi, ST. MT., Ph.D**
NIP : 197210012003121002

(Pembimbing 1)

2. **Dr. Ir. Wirawan, DEA**
NIP : 196311091989031011

(Pembimbing 2)

3. **Dr. Ir. Titiek Suryani, MT.**
NIP : 196411301989032001

(Penguji 1)

4. **Dr. Istas Pratomo, ST., MT**
NIP : 197903252003121001

(Penguji 2)

5. **Dr. Ir. Achmad Affandi, DEA**
NIP : 196510141990021001

(Penguji 3)

Direktur Program Pascasarjana,

Prof. Dr. Ir. Adi Soeprijanto, MT

NIP. 196404051990021001

Tesis telah disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (MT)

Di
Institut Teknologi Sepuluh Nopember

oleh :
Faridatun Nadziroh
NRP. 2213203018

Tanggal Ujian : 17 Juni 2015

Periode Wisuda : September 2015

Disetujui oleh:

1. **Eko Setidjadi, ST. MT., Ph.D**
NIP : 197210012003121002

(Pembimbing 1)

2. **Dr. Ir. Wirawan, DEA**
NIP : 196311091989031011

(Pembimbing 2)

3. **Dr. Ir. Titiek Suryani, MT.**
NIP : 196411301989032001

(Penguji 1)

4. **Dr. Istas Pratomo, ST., MT**
NIP : 197903252003121001

(Penguji 2)

5. **Dr. Ir. Achmad Affandi, DEA**
NIP : 196510141990021001

(Penguji 3)

Direktur Program Pascasarjana,

Prof. Dr. Ir. Adi Soeprijanto, MT

NIP. 196404051990021001

DESAIN DAN ANALISA MANAJEMEN KONSUMSI DAYA PADA WSN UNTUK SISTEM MONITORING KESEHATAN STRUKTUR (SMKS) JEMBATAN

Nama Mahasiswa : Faridatun Nadziroh
NRP : 2213203018
Pembimbing : Eko Setijadi, ST., MT., Ph.D
: Dr. Ir. Wirawan, DEA

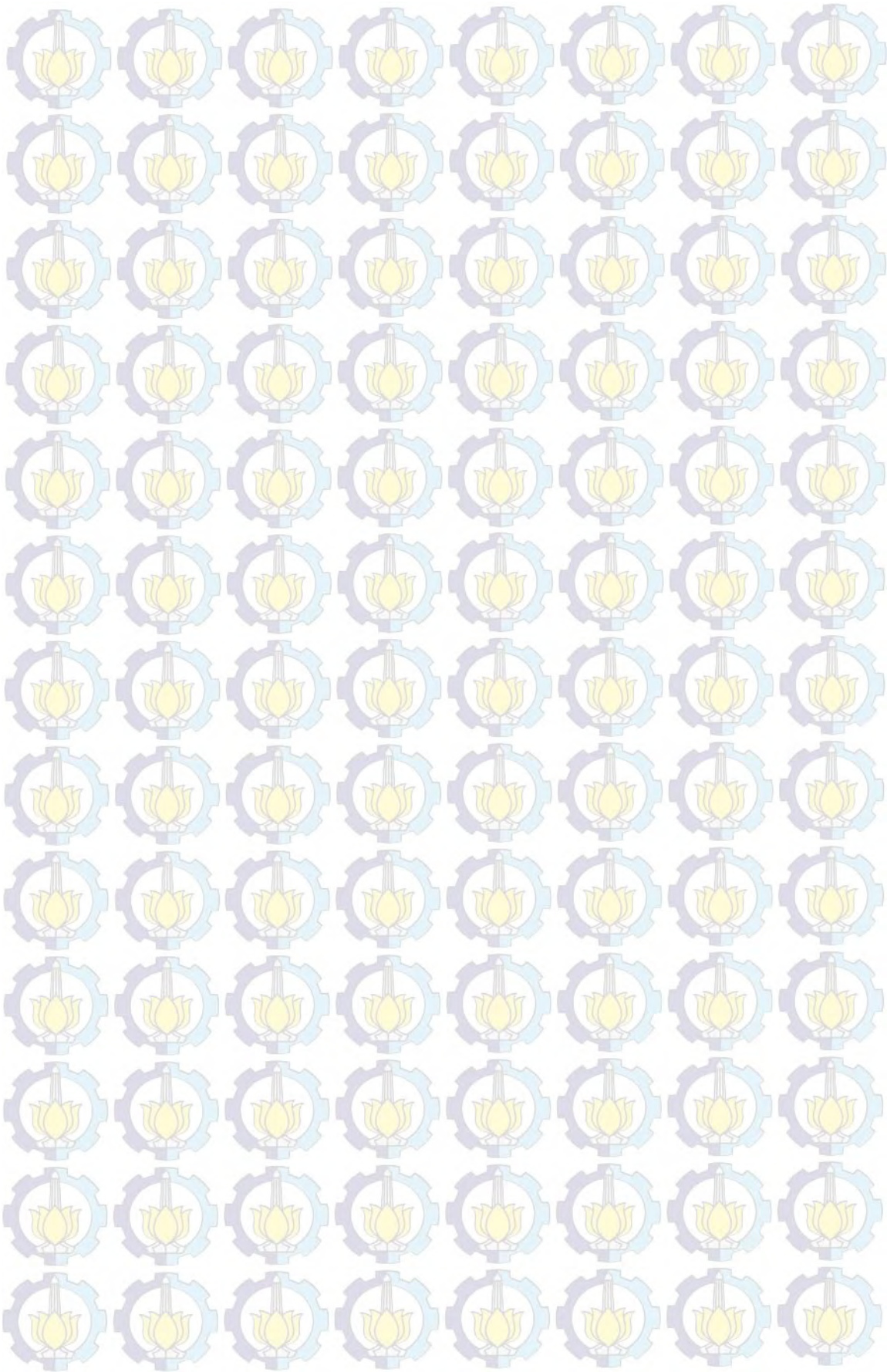
ABSTRAK

Peristiwa robohnya jembatan Kutai Kertanegara pada tanggal 26 November 2011 yang memiliki panjang 720 meter mengakibatkan banyak korban. Untuk mengatasi hal serupa terjadi, diperlukan adanya suatu sistem pemantauan terhadap jembatan yang berguna untuk mengetahui lebih dini tentang kerusakan pada jembatan sehingga dapat meminimalisir kerusakan yang lebih besar. Pemantauan kesehatan struktur jembatan dapat dilakukan dengan mengimplementasikan Sistem Monitoring Kesehatan Stuktur (SMKS) Jembatan. SMKS memanfaatkan sensor-sensor dalam pemantauan dengan model komunikasi menggunakan sistem wireless atau bisa di sebut *Wireless Sensor Network* (WSN).

Dalam bekerja WSN memerlukan daya yang dicatu oleh baterai. Akan tetapi WSN memiliki kendala yakni kapasitas dan daya baterai kecil karena WSN dicatu oleh baterai yang mempunyai *lifetime* sangat terbatas. Mengatasi hal itu, diperlukan adanya suatu cara untuk memanajemen konsumsi daya pada sensor. Teknik manajemen konsumsi daya di munculkan dengan mengatur kondisi *sleep* atau *awake* pada node sensor serta mendesain topologi serta routing yang digunakan. Pemilihan topologi dan routing yang tepat yakni dengan mempertimbangkan parameter energi, jarak, *packet loss*, *throughput* serta *delay* dari sumber ke tujuan dapat menjadikan proses transmisi lebih maksimal. Selanjutnya membandingkan hasil kinerja dari topologi dan routing yang digunakan. Topologi yang digunakan yakni topologi star, tree dan mesh. Sedangkan untuk routingsnya menggunakan routing AODV dan DSDV. Penelitian ini berbasis simulasi dengan menggunakan Network Simulator-2 (NS-2).

Dari hasil analisa proses manajemen energi pada sistem monitoring struktur jembatan dapat di lakukan dengan memilih topologi yang terbaik dengan routing yang baikpula. Topologi yang terbaik yang dapat diterapkan pada Sistem Monitoring Kesehatan Struktur Jembatan adalah topologi *mesh* dengan routing AODV.

Kata Kunci : SMKS, WSN, Routing, Topologi



DESIGN AND ANALYSIS OF POWER CONSUMPTION MANAGEMENT WSN FOR BRIDGE STRUCTURAL HEALTH MONITORING SYSTEM (SHMS)

Student Name : Faridatun Nadziroh
NRP : 2213203018
Supervisor : Eko Setijadi, ST., MT., Ph.D.
Dr. Ir. Wirawan, DEA

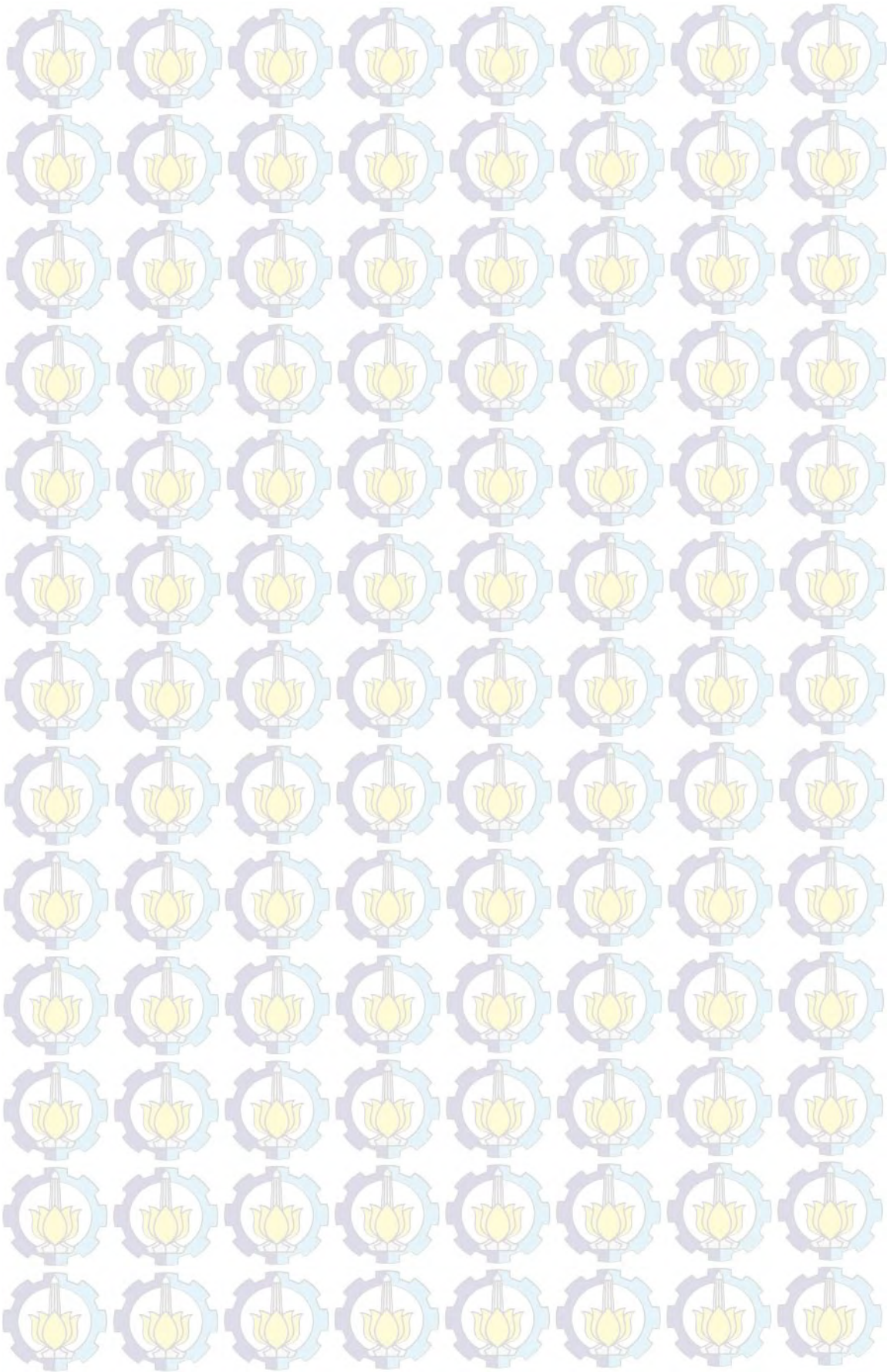
ABSTRACT

Kutai Kartanegara bridge collapse incident on November 26, 2011, which has a length of 720 meters resulted in many casualties. To overcome similar things happen, needed a monitoring system for bridges that are useful to know earlier about the damage to the bridge so as to minimize the damage is greater. Bridge structural health monitoring can be done by implementing the Structure Health Monitoring System (SHMS) Bridge. SHMS utilizes the sensors in monitoring communication model using wireless system can be called Wireless Sensor Network (WSN).

In working WSN require power supplied by the battery. However, WSN have the capacity constraints and the small battery power because WSN supplied by batteries which have a very limited lifetime. Overcoming it, needed a way to manage power consumption on the sensor. The power consumption management techniques appear to set the conditions of sleep or awake at the sensor node as well as designing the topology and routing are used. Selection of the appropriate routing topology and the taking into account of energy parameters, distance, packet loss, throughput and delay from source to destination can make the transmission process more leverage. Furthermore, comparing the results of the performance of topology and routing are used. Topology used the star topology, tree and mesh. As for routing using the routing AODV and DSDV. Bridges in real time.

From the analysis of the energy management process monitoring system bridge structure can be done by selecting the best topology with routing baik pula. Topology is the best that can be applied to the Bridge Structure Health Monitoring System is a mesh topology with AODV routing.

Keywords: SMKS, WSN, Routing, Topology



DESIGN AND ANALYSIS OF POWER CONSUMPTION MANAGEMENT WSN FOR BRIDGE STRUCTURAL HEALTH MONITORING SYSTEM (SHMS)

Student Name : Faridatun Nadziroh
NRP : 2213203018
Supervisor : Eko Setijadi, ST., MT., Ph.D.
Dr. Ir. Wirawan, DEA

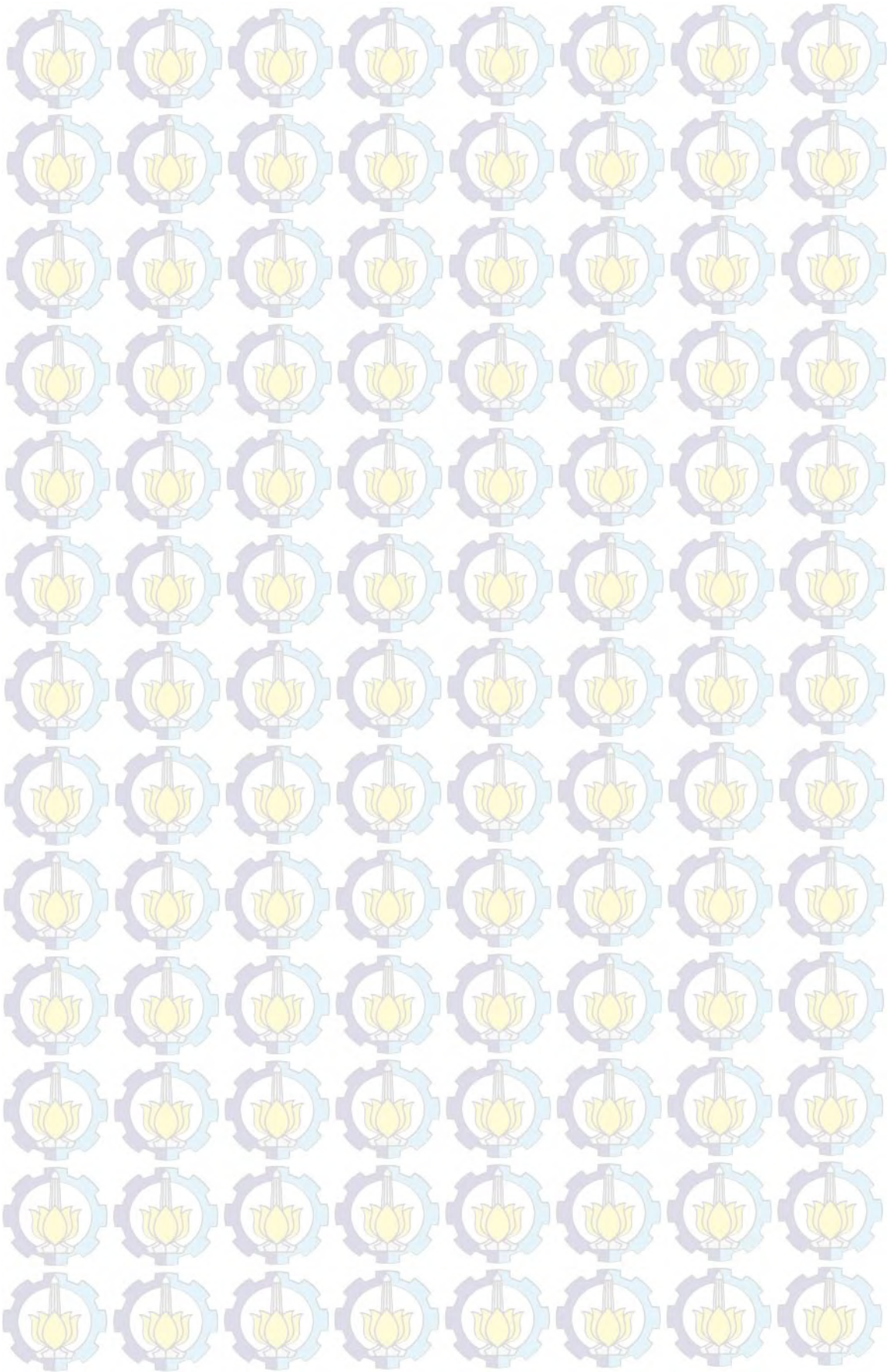
ABSTRACT

Kutai Kartanegara bridge collapse incident on November 26, 2011, which has a length of 720 meters resulted in many casualties. To overcome similar things happen, needed a monitoring system for bridges that are useful to know earlier about the damage to the bridge so as to minimize the damage is greater. Bridge structural health monitoring can be done by implementing the Structure Health Monitoring System (SHMS) Bridge. SHMS utilizes the sensors in monitoring communication model using wireless system can be called Wireless Sensor Network (WSN).

In working WSN require power supplied by the battery. However, WSN have the capacity constraints and the small battery power because WSN supplied by batteries which have a very limited lifetime. Overcoming it, needed a way to manage power consumption on the sensor. The power consumption management techniques appear to set the conditions of sleep or awake at the sensor node as well as designing the topology and routing are used. Selection of the appropriate routing topology and the taking into account of energy parameters, distance, packet loss, throughput and delay from source to destination can make the transmission process more leverage. Furthermore, comparing the results of the performance of topology and routing are used. Topology used the star topology, tree and mesh. As for routing using the routing AODV and DSDV. Bridges in real time.

From the analysis of the energy management process monitoring system bridge structure can be done by selecting the best topology with routing baik pula. Topology is the best that can be applied to the Bridge Structure Health Monitoring System is a mesh topology with AODV routing.

Keywords: SMKS, WSN, Routing, Topology



KATA PENGANTAR

Dengan Nama Allah Yang Maha Pengasih lagi Maha Penyayang.

Segala puji syukur kepada Allah SWT. atas segala rahmat dan karunia yang telah dilimpahkan, sehingga penulisan tesis dengan judul :

“Desain Dan Analisa Manajemen Konsumsi Daya Pada WSN Untuk Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan”

dapat diselesaikan dengan baik. Buku tesis ini disusun untuk memenuhi salah satu syarat memperoleh gelar Magister pada Program Studi Teknik Elektro, Bidang Keahlian Telekomunikasi Multimedia, Institut Teknologi Sepuluh Nopember.

Pada kesempatan ini penulis ingin menyampaikan terima kasih kepada :

1. Orang tua penulis, Bapak Ir. R. Sofwan Aziz dan Ibu Dra. Masalah, yang senantiasa mendo'akan, memberikan nasihat, memotivasi dan membimbing penulis dengan penuh kesabaran. Merekalah yang menjadi semangat penulis untuk terus berusaha menjadi manusia yang bermanfaat.
2. Bapak Eko Setijadi, ST.,MT., Ph.D. dan Bapak Dr. Ir. Wirawan, DEA. selaku dosen pembimbing, terima kasih atas bimbingan, kesabaran dan pendorong semangat dalam menyelesaikan tesis ini.
3. Bapak dan Ibu dosen S2, terima kasih atas bimbingan dan ilmu pengetahuan yang diberikan selama kuliah.
4. ITS dan DIKTI yang telah memberikan kesempatan pada penulis untuk mendapatkan pendidikan S2 dalam bentuk beasiswa *fresh graduate*.
5. Seluruh keluarga, saudara serta kerabat yang telah banyak memberi dukungan.
6. Evy Nur Amalina, yang senantiasa memberikan masukan, ide, motivasi dan dukungan dalam menyelesaikan tesis ini
7. Shazana D. A. dan Rekan-rekan Tim SHM (*Structural Health Monitoring*) Jembatan, terima kasih atas dukungan, bantuan dan kerja samanya.
8. Teman-teman TMM 2013 seperjuangan yang selalu memberi semangat.

Semoga Allah SWT. membalas segala budi baik yang telah diberikan dalam penyusunan tesis ini sehingga dapat terselesaikan dengan baik.

Penulis menyadari bahwa dalam penulisan tesis ini masih jauh dari sempurna, untuk itu demi perbaikan dan penyempurnaan tesis, maka kritik dan saran sangat diharapkan. Besar harapan penulis bahwa buku tesis ini dapat memberikan informasi dan manfaat bagi pembaca pada umumnya dan mahasiswa Jurusan Teknik Elektro pada khususnya.

Surabaya, Juni 2015

Penulis

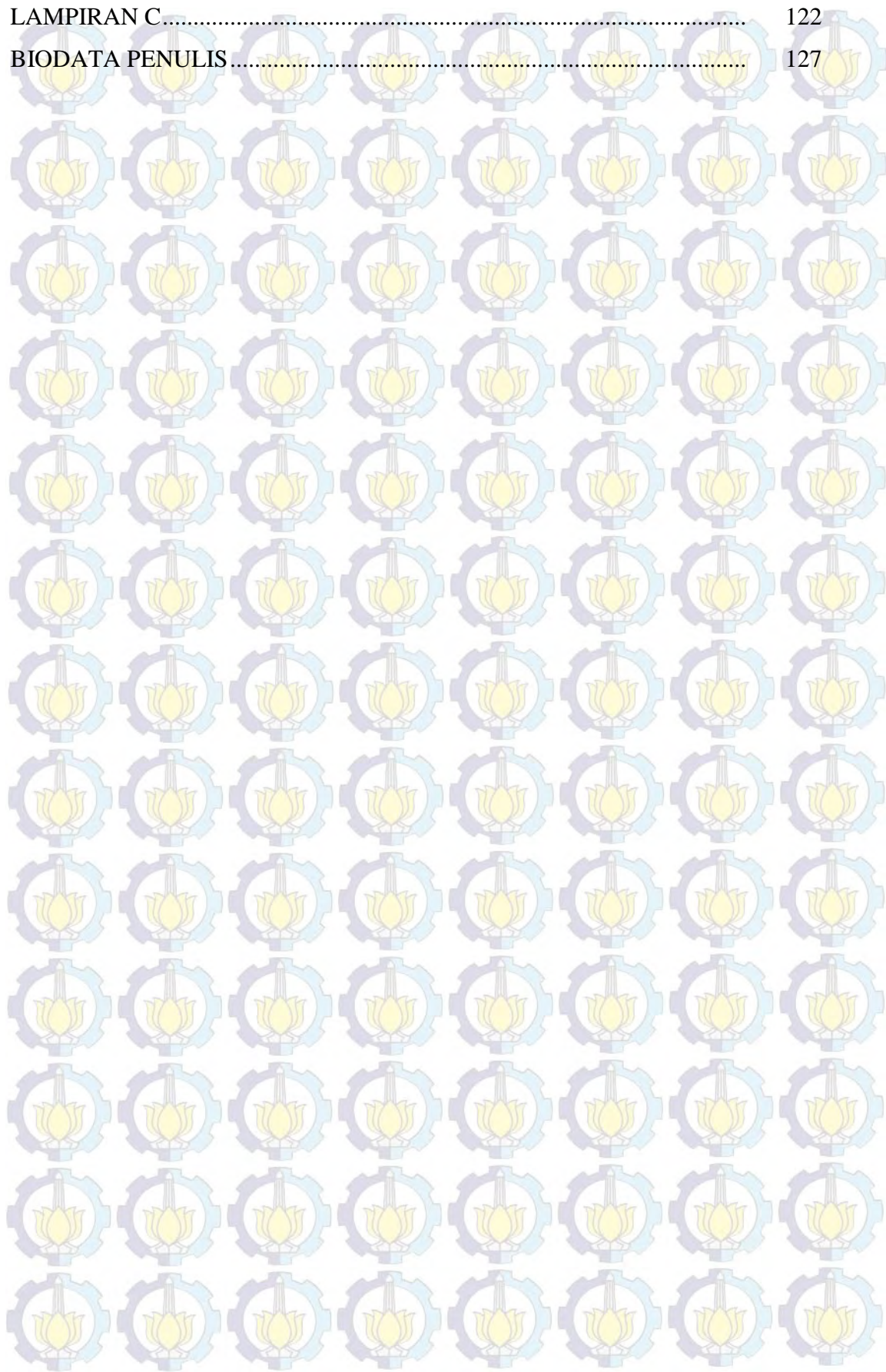
DAFTAR ISI

	HALAMAN
HALAMAN JUDUL	i
PERNYATAAN KEASLIHAN TESIS	ii
HALAMAN PENGESAHAN	iii
ABSTRAK	v
ABSTRACT	vii
KATA PENGANTAR	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xv
DAFTAR TABEL	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah	5
1.4 Tujuan	5
1.5 Manfaat	5
1.6 Sistematika Penulisan	5
BAB 2 TINJAUAN PUSTAKA	7
2.1 Sistem Monitoring Kesehatan Stuktur (SMKS)	7
2.1.1 SMKS Jembatan	7
2.1.2 Komponen Sistem Monitoring	8
2.1.3 Syarat Moniroting Jembatan	9
2.1.4 Klasifikasi Moniroting	10
2.1.5 Parameter Akuisi Data Dalam Monitoring Jembatan	11
2.2 <i>Wireless Sensor Network</i> (WSN)	12
2.2.1 Karakteristik WSN	14
2.2.2 Topologi Jaringan WSN	15

2.3 DSDV	18
2.4 AODV.....	20
2.5 Zigbee (802.15.4)	22
2.5.1 Karakteristik Zigbee	22
2.5.2 Arsitektur Zigbee.....	23
2.5.3 Xbee Pro versi v1.xCx.....	24
2.6 Arduino Due	25
2.7 Sensor	26
2.7.1 Sensor Accelerometer.....	27
2.8 Standar IEEE 802.15.4	27
2.9 Network Simulator-2 (NS-2).....	27
2.7.1 Komponen Pembangun NS2	28
BAB 3 METODE PENELITIAN	30
3.1 Rancangan Penelitian	29
3.2 Manajemen Sumber Daya	33
3.3 Metode Pengiriman Data.....	34
3.4 Pemodelan Sistem	33
3.4.1 Transmit Power Output	36
3.4.2 Receive Threshold.....	36
3.4.3 Transmit Power	36
3.4.4 Receive Power.....	37
3.4.5 Initial Energy	38
3.2 Desain Sensor pada Jembatan	38
3.2.1 Jembatan SURAMADU	38
3.2.2 Desain Jembatan Dalam Simulasi	40
3.2.3 Desain Peletakan Sensor.....	41
3.3 Desain Topologi	44
3.3.1 Topologi Star.....	44
3.3.2 Topologi Mesh.....	45
3.3.3 Topologi Tree	46
3.4 Routing Protokol	47

3.4.1 AODV	48
3.4.2 DSDV	49
3.5 Model Simulasi	50
3.5.1 Format Trace File	51
3.6 Parameter Kinerja Simulasi	54
3.7 Parameter Simulasi	56
3.8 Validasi Parameter Simulasi	56
3.8.1 Perangkat Validasi	57
BAB 4 HASIL DAN ANALISA	61
4.1 Skenario Pengujian Sistem	61
4.2 Asumsi Pertama	62
4.2.1 Topologi Star	64
4.2.2 Topologi Mesh	67
4.2.3 Topologi Tree	70
4.3 Asumsi Kedua	74
4.3.1 Topologi Star	76
4.3.2 Topologi Mesh	79
4.3.3 Topologi Tree	82
4.4 Analisa	85
4.5 Validasi Parameter Simulasi	86
4.6 Pengujian Kalibrasi Node	87
4.7 Manajemen Transmisi Data	87
4.8 Analisa Validasi Parameter Simulasi	90
BAB 5 KESIMPULAN DAN SARAN	95
5.1 Kesimpulan	95
5.2 Saran	97
DAFTAR PUSTAKA	99
LAMPIRAN A	103
LAMPIRAN B	121

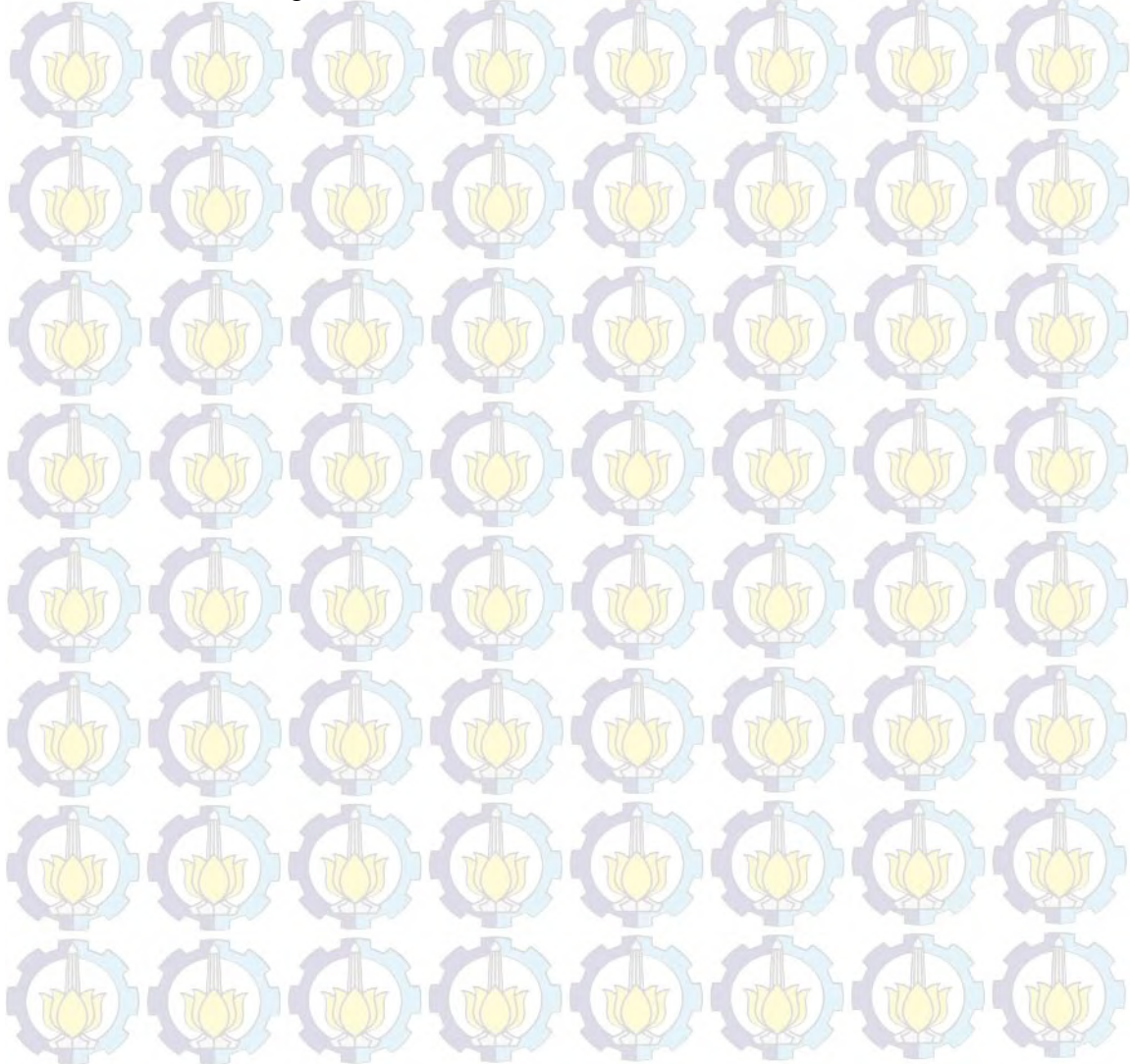
LAMPIRAN C.....	122
BIODATA PENULIS.....	127



DAFTAR TABEL

	HALAMAN
Tabel 2.1. Update Rute pada Node A	27
Tabel 2.2. Spesifikasi Modul XBee Pro versi v1	25
Tabel 2.2. Spesifikasi Arduino Due	26
Tabel 3.1. Parameter Simulasi	56
Tabel 4.1. Hasil Simulasi Topologi Star Dengan Routing DSDV Dengan Asumsi Pertama	65
Tabel 4.2. Hasil Simulasi Topologi Star Dengan Routing AODV Dengan Asumsi Pertama	65
Tabel 4.3. Hasil Simulasi Topologi Mesh Dengan Routing DSDV Dengan Asumsi Pertama	68
Tabel 4.4. Hasil Simulasi Topologi Mesh Dengan Routing AODV Dengan Asumsi Pertama	68
Tabel 4.5. Hasil Simulasi Topologi Tree Dengan Routing DSDV Dengan Asumsi Pertama	71
Tabel 4.6. Hasil Simulasi Topologi Tree Dengan Routing AODV Dengan Asumsi Pertama	71
Tabel 4.7. Tabel Gabungan Hasil Simulasi Topologi Star, Mesh Dan Tree Asumsi Pertama Routing AODV	73
Tabel 4.8. Tabel Gabungan Hasil Simulasi Topologi Star, Mesh Dan Tree Asumsi Pertama Routing DSDV	73
Tabel 4.9. Hasil Simulasi Topologi Star Dengan Routing DSDV Dengan Asumsi Kedua	77
Tabel 4.10. Hasil Simulasi Topologi Star Dengan Routing AODV Dengan Asumsi Kedua	77
Tabel 4.11. Hasil Simulasi Topologi Mesh Dengan Routing DSDV Dengan Asumsi Kedua	80
Tabel 4.12. Hasil Simulasi Topologi Mesh Dengan Routing AODV	

Dengan Asumsi Kedua.....	80
Tabel 4.13. Hasil Simulasi Topologi Tree Dengan Routing DSDV	
Dengan Asumsi Kedua.....	82
Tabel 4.14. Hasil Simulasi Topologi Tree Dengan Routing AODV	
Dengan Asumsi Kedua.....	83
Tabel 4.15. Tabel Gabungan Hasil Simulasi Topologi Star, Mesh Dan Tree	
Asumsi Kedua Routing AODV.....	85
Tabel 4.16. Tabel Gabungan Hasil Simulasi Topologi Star, Mesh Dan Tree	
Asumsi Kedua Routing DSDV	85
Tabel 4.17. Perbandingan Konsumsi Arus Berdasarkan Kondisi Node.....	90
Tabel 4.18. Hasil Pengukuran <i>Throughput</i>	90
Tabel 4.19. Hasil Pengukuran <i>Delay</i>	92
Tabel 4.20. Hasil Pengukuran Konsumsi Arus.....	92



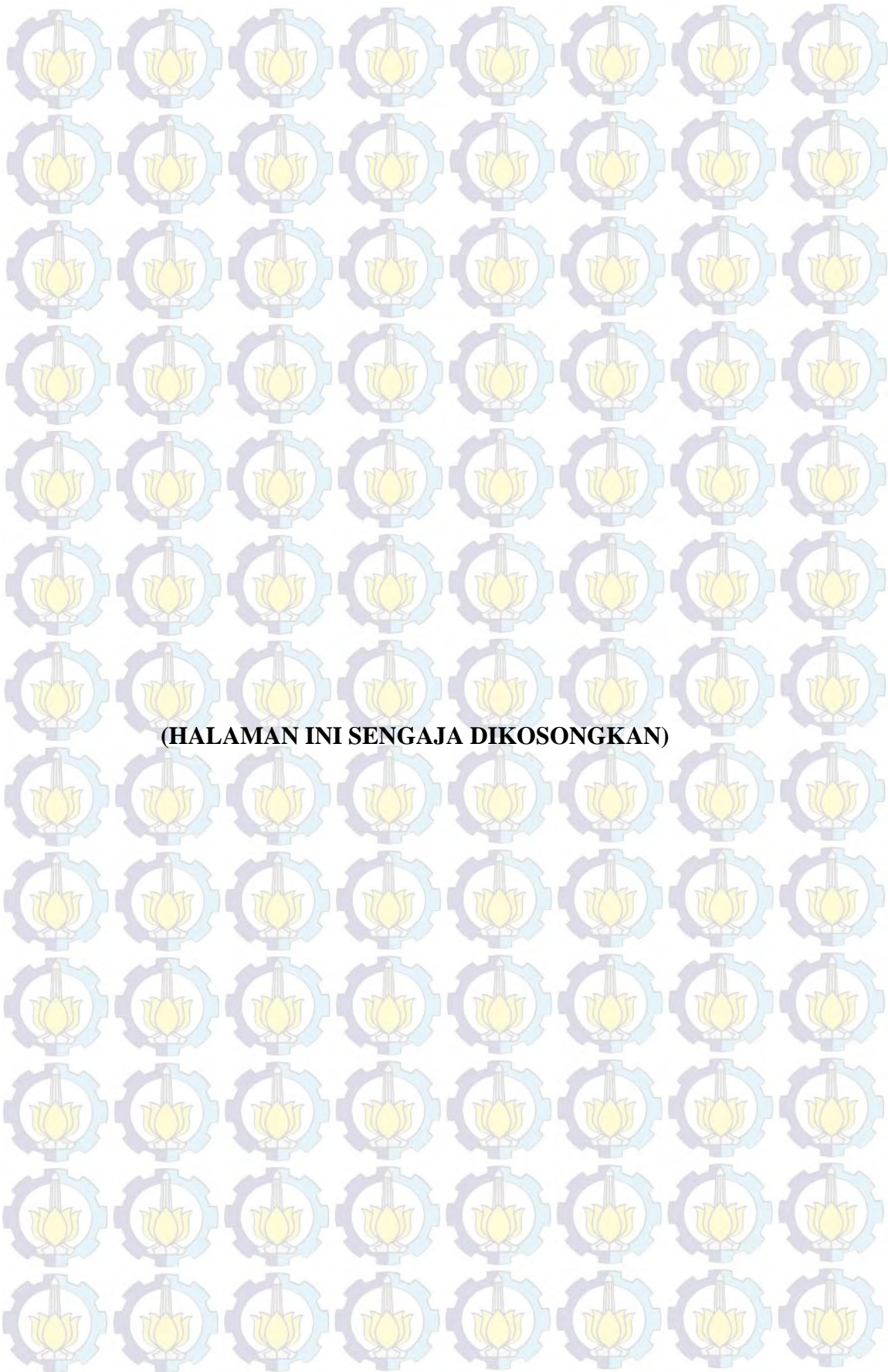
DAFTAR GAMBAR

HALAMAN

Gambar 2.1. Klasifikasi sistem Monitoring Berdasarkan Kemampuan Pendeteksian dan Pengambilan Keputusan.....	11
Gambar 2.2. Arsitektur WSN Secara Umum	13
Gambar 2.3. Contoh Aplikasi WSN (a) WSN pada long span bridge, (b)WSN pada Road Bridge.....	14
Gambar 2.4. Topologi Star	15
Gambar 2.5. Topologi Tree	16
Gambar 2.6. Topologi Mesh.....	16
Gambar 2.7. Topologi Mesh dengan 2 <i>coordinator</i> dan <i>sink</i>	17
Gambar 2.8. Topologi Mesh terdapat <i>coordinator</i> yang <i>down</i>	17
Gambar 2.9. Pembuatan Rute Sementara Node A (a) Link Dari Node A ke B Breaks, (b) Node A Menyebarkan ROUTE-REQUEST ke Node Tetangga (c) Node A Memilih Node E Sebagai Hop Selanjutnya,	20
Gambar 2.10. Mekanisme Penemuan Rute (a) Node sumber S mengirim RREQ ke node tetangganya, (b) Node tujuan mengirimkan RREP kembali ke sumber	21
Gambar 2.11. Perangkat Zigbee.....	23
Gambar 2.12. Arsitektur ZigBee.....	26
Gambar 2.13. Arduino Due.....	27
Gambar 2.14. Komponen Pembangun NS2.....	28
Gambar 3.1. Road Map Penelitian	31
Gambar 3.2. Flowchart Rencana Penelitian	33
Gambar 3.3. Struktur Jembatan Suramadu.....	39
Gambar 3.4. Jembatan Suramadu a) SURAMADU Tampak Atas b) SURAMADU Tampak Samping.....	40

Gambar 3.5. Desain Jembatan Dalam Simulasi	
a) Desain Jembatan Dalam Simulasi Tampak Atas	
b) Desain Jembatan Dalam Simulasi Tampak Samping	41
Gambar 3.6. Desain Peletakan Sensor Asumsi Pertama.....	43
Gambar 3.7. Desain Peletakan Sensor Asumsi Kedua	43
Gambar 3.8. Desain Topologi Star Asumsi Pertama	44
Gambar 3.9. Desain Topologi Star Asumsi Kedua.....	45
Gambar 3.10. Desain Topologi Mesh Asumsi Pertama.....	45
Gambar 3.11. Desain Topologi Mesh Asumsi Kedua	46
Gambar 3.12. Desain Topologi Tree Asumsi Pertama	46
Gambar 3.13. Desain Topologi Tree Asumsi Kedua.....	47
Gambar 3.14. Paket <i>processing</i> AODV	48
Gambar 3.15. Paket <i>processing</i> DSDV.....	49
Gambar 3.16. Proses simulasi pada NS2	51
Gambar 3.17. Format Trace File.....	51
Gambar 3.18. Contoh Trace File Pada Topologi Mesh Asumsi Pertama.....	53
Gambar 3.19. Desain Implementasi Sistem untuk Validasi Parameter	
Simulasi dengan Jarak 100 Meter	57
Gambar 3.20. Implementasi Sistem Untuk Validasi Parameter Simulasi	57
Gambar 3.21. Bentuk Fisik Node	58
Gambar 3.22. Susunan Node Koordinator	59
Gambar 3.23. Susunan Node Sensor.....	59
Gambar 4.1. Pembagian Arah Jalur Pada Jembatan	61
Gambar 4.2. Skenario Pada Asumsi Pertama	63
Gambar 4.3. Topologi Star Asumsi Pertama	64
Gambar 4.4. Energi End-To-End Topologi Star Asumsi Pertama Routing	
DSDV	66
Gambar 4.5. Energi End-To-End Topologi Star Asumsi Pertama Routing	
AODV	67
Gambar 4.6. Topologi Mesh Asumsi Pertama.....	67

Gambar 4.7. Energi End-To-End Topologi Mesh Asumsi Pertama Routing	
DSDV	69
Gambar 4.8. Energi End-To-End Topologi Mesh Asumsi Pertama Routing	
AODV	70
Gambar 4.9. Topologi Tree Asumsi Pertama	70
Gambar 4.10. Energi End-To-End Topologi Tree Asumsi Pertama Routing	
DSDV	72
Gambar 4.11. Energi End-To-End Topologi Tree Asumsi Pertama Routing	
AODV	72
Gambar 4.12. Skenario Pada Asumsi Kedua	75
Gambar 4.13. Topologi Star Asumsi Kedua	76
Gambar 4.14. Energi End-To-End Topologi Star Asumsi Kedua Routing	
DSDV	77
Gambar 4.15. Energi End-To-End Topologi Star Asumsi Kedua Routing	
AODV	78
Gambar 4.16. Topologi Mesh Asumsi Kedua	79
Gambar 4.17. Energi End-To-End Topologi Mesh Asumsi Kedua Routing	
DSDV	80
Gambar 4.18. Energi End-To-End Topologi Mesh Asumsi Kedua Routing	
AODV	81
Gambar 4.19. Topologi Tree Asumsi Kedua	82
Gambar 4.20. Energi End-To-End Topologi Tree Asumsi Kedua Routing	
DSDV	83
Gambar 4.21. Energi End-To-End Topologi Tree Asumsi Kedua Routing	
AODV	84
Gambar 4.22. Proses Pengambilan Data Secara Real	86
Gambar 4.23. Tampilan Pemograman Arduino	87
Gambar 4.24. <i>Window Serial Monitoring</i> Pada Node Koordinator	87
Gambar 4.25. Format Protokol Pemaketan Data pada Node Sensor.....	88
Gambar 4.26. Waktu Pengiriman Data	89
Gambar 4.27. Perbandingan <i>Throughput</i> Validasi.....	91
Gambar 4.27. Perbandingan <i>Delay</i> Validasi	92



(HALAMAN INI SENGAJA DIKOSONGKAN)

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Beberapa bangunan memiliki kerentanan terhadap kerusakan, sebagai contoh jembatan, bendungan, bandara dan stadion. Kerusakan ini tidak mudah untuk diantisipasi sehingga terkadang dapat menimbulkan korban. Pada tahun 2011, telah terjadi peristiwa robohnya sebuah jembatan di Indonesia, yakni Jembatan Kutai Kartanegara yang memiliki panjang 720 meter dan mengakibatkan banyak korban [1]. Peristiwa runtuhnya jembatan ini disebabkan oleh adanya tegangan yang berlebihan terhadap jembatan tersebut.

Kerusakan suatu jembatan umumnya disebabkan oleh beberapa faktor. Faktor ini dapat dikerucutkan menjadi 2 tipe yakni faktor internal dan faktor eksternal [2]. Faktor internal meliputi kerusakan yang disebabkan oleh komponen penyusun dari jembatan tersebut. Sedangkan faktor eksternal meliputi kerusakan yang disebabkan oleh alam sekitar.

Untuk meminimalisir terjadinya kerusakan jembatan maka perlu diterapkan suatu sistem dan teknologi pemantauan. Teknologi ini adalah Bridges Structural Health Monitoring System atau bisa disebut Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan. Teknologi SMKS ini memiliki potensi yang sangat baik untuk meningkatkan operasi rutin dan pemeliharaan struktur pada jembatan [2]. Teknologi ini dapat memperpanjang usia dari jembatan karena dapat mendeteksi lebih awal. Teknologi ini menggunakan sensor-sensor. Adapun sensor-sensor yang digunakan diantaranya sensor gerakan dan getaran (accelerometer), sensor tekanan (pressure gauge), sensor suhu (thermocouple), dan sensor akustik (piezoelectric) [3]. Sensor-sensor tersebut bertugas untuk mengambil data sesuai dengan jenis sensor, kemudian data dari sensor tersebut akan dikirim menuju pusat, sehingga pusat akan mengetahui kondisi terkini dari jembatan tersebut.

Pada SMKS terdapat beberapa tingkatan dalam penilaian kondisi jembatan yaitu identifikasi anomali dan kerusakan dalam struktur, lokalisasi kerusakan, kuantifikasi keparahan kerusakan, dan prediksi sisa umur layanan struktur [2]. Tingkatan ini bertujuan untuk melakukan pencegahan dan mendeteksi dengan cepat masalah yang ditemukan.

Untuk proses komunikasi yang digunakan pada SMKS di bagi menjadi 2 jenis yakni komunikasi berbasis kabel dan nirkabel. Komunikasi berbasis kabel memiliki kelebihan yaitu data yang dikirimkan dari sensor sangat akurat dan juga tidak memerlukan daya yang cukup banyak karena dayanya diambil langsung dari pusat. Akan tetapi komunikasi berbasis kabel ini terkendala pada instalasi kabel sehingga semakin jauh jarak jangkauannya maka memerlukan kabel yang cukup banyak yang juga berpengaruh pada biaya instalasinya. Menurut [4] sekitar 25 persen dari total biaya anggaran SMKS digunakan untuk instalasi kabel. Meninjau hal ini, maka perlu adanya sistem monitoring yang tidak memerlukan instalasi kabel yakni menggunakan jaringan sensor nirkabel [5]. Komunikasi nirkabel tersebut atau lebih dikenal dengan wireless sensor network (WSN). WSN mempunyai kelebihan yaitu tidak memerlukan komponen tambahan seperti saluran kabel, sensor mudah diganti jika mengalami kerusakan, mudah dikonfigurasi ulang, dan dengan sistem Ad-Hoc dan Multi-Hop komunikasi data menjadi lebih praktis.

Dalam pemantauan kesehatan jembatan yang menggunakan WSN pasti memerlukan konsumsi daya agar sensor-sensor dapat bekerja dengan baik. Namun juga perlu di pertimbangkan akan daya yang digunakan, dimana perlu adanya teknik atau metode yang digunakan agar konsumsi daya yang digunakan lebih efisien. Sedangkan WSN memiliki keterbatasan daya dimana rata-rata sensor dicatu oleh baterai yang mempunyai *lifetime* sangat terbatas. Untuk peralatan komunikasi yang digunakan untuk mengirimkan data melalui jaringan nirkabel harus memperhatikan ketersediaan bandwidth, range sensor dan range komunikasi sensor [6]. Pada penelitian [7] terdapat beberapa teknik untuk menghemat daya secara umum, yakni:

1. Penjadwalan node

2. Mengatur control daya dengan menyesuaikan jangkauan transmisi di sekitar node

3. Mengurangi jumlah data yang ditransmisikan dan menghindari aktivitas yang sia-sia.

Selain itu langkah penghematan daya pada WSN bertujuan agar masa hidup sensor lebih lama sehingga penggunaan sensor dapat lebih maksimal dan dapat digunakan untuk proses monitoring secara kontinyu. Pada [8] mengusulkan penghematan daya dengan mengatur topologi yang digunakan dengan menfokuskan pada topologi cluster-tree dengan berbasis simulasi. Daya yang digunakan lebih efisien karena topologi disesuaikan dengan lingkungan yang akan diamati, namun studi ini tidak mempertimbangkan konteks implementasi jaringan secara *real time*.

Pada penelitian ini, teknik manajemen konsumsi daya didapatkan melalui pengaturan node-node sensor dengan memanfaatkan kondisi *sleep/awake* pada node sensor tersebut. Node *awake* ketika terjadi proses pentransmisian data dengan node tetangganya. Namun ketika node tetangga mengalami penurunan daya, maka node yang sedang mentransmisikan data akan mencari node tetangga lain yang masih memiliki daya yang cukup banyak. Sedangkan untuk node yang tidak mentransmisikan data dalam kondisi *sleep*. Pemilihan node dalam proses pentransmisian data mempertimbangkan energi, jarak, *packet loss*, *throughput* serta *delay* pada node yang akan dituju.

Dalam pemilihan node untuk proses pentransmisian data dipengaruhi oleh rute atau jalur yang akan dilalui. Rute yang tepat akan menjadikan proses transmisi lebih maksimal. Node-node yang dipilih dalam proses transmisi berperan tidak hanya menjadi pengirim dan penerima, namun juga berperan sebagai penunjang node yang lainnya. Maka dari itu diperlukan routing protokol untuk menunjang proses pentransmisian antar node-node. Pada jaringan sensor nirkabel memanfaatkan sistem Ad-Hoc dan MultiHop.

Pada sistem Ad-hoc terdapat jenis routing protokol diantaranya AODV dan DSDV. Beberapa penelitian telah melakukan pengamatan tentang routing protokol untuk jaringan ad-hoc. Pada studi [9] melakukan penelitian tentang routing protokol AODV dengan memanfaatkan trafik CBR dan TCP, dimana

untuk pengiriman paket menggunakan Packet Delivery Fraction (PDF). Pada TCP terdapat proses pengiriman ulang untuk paket yang mengalami error sedangkan CBR tidak ada pengiriman ulang sehingga banyaknya paket yang diterima lebih kecil karena adanya kemungkinan error. Menurut studi [10] kinerja routing protokol DSDV (Destination Sequenced Distance Vector) pada jaringan wireless dipengaruhi oleh lama waktu dan kecepatan, dimana ketika semakin lama waktu dan semakin sedikit kecepatan maka kinerja DSDV semakin tinggi. Penelitian ini berbasis simulasi menggunakan Network Simulator-2 (NS2). Network simulator ini merupakan salah satu simulator berbasis open source yang digunakan untuk membantu analisa dalam hal pemodelan media, protocol dan trafiknya.

Penelitian ini berfokus pada desain dan analisa untuk manajemen konsumsi daya dengan cara mengkondisikan sleep/awake sensor pada node-node melalui pengaturan topologi serta routing protokolnya. Sehingga hasil simulasi dapat diimplementasikan pada Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.

1.2. Rumusan Masalah

Pada penelitian ini, rumusan permasalahan yang diambil adalah :

1. Bagaimana manajemen daya pada node untuk mendapatkan konsumsi daya yang lebih efisien.
2. Routing manakah yang baik untuk diterapkan pada Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.
3. Topologi apa yang sesuai untuk di terapkan pada Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.

1.3. Batasan Masalah

Pada penelitian ini, batasan masalah yang diambil adalah :

1. Simulasi menggunakan NS-2.
2. Topologi yang digunakan adalah star, tree dan mesh
3. Wireless module yang digunakan adalah Protocol 802.15.4 dan mengacu pada datasheet XBee Pro versi v1.
4. Routing protokol menggunakan DSDV dan AODV.

5. Sensor yang digunakan adalah sensor sensor accelerometer
6. Protocol 802.15.4 memiliki coverage area 250 m.

1.4. Tujuan

Tujuan dari penelitian ini adalah :

- 1) Memperoleh nilai throughput, delay, paket dan energi untuk setiap node pada saat node berkomunikasi.
- 2) Mendapatkan topologi dan routing terbaik untuk diterapkan pada Sistem Monitoring Struktur (SMKS) Jembatan.

1.5. Manfaat

Dari usulan penelitian tesis ini, diharapkan dapat memberikan kontribusi dalam hal manajemen konsumsi daya pada WSN untuk Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.

1.6. Sistematika Penulisan

Untuk memudahkan pembahasan, usulan tesis ini disusun dengan sistematika sebagai berikut :

BAB 1. PENDAHULUAN

Bagian pendahuluan berisi latar belakang, rumusan masalah, batasan masalah, metode penelitian yang digunakan, dan sistematika penulisan. Bagian ini merupakan hal-hal yang mendasari dan pentingnya penelitian yang dilakukan.

BAB 2. TINJAUAN PUSTAKA

Pada bab ini berisi tinjauan pustaka yang mendukung dan berkaitan dengan penelitian yaitu tentang Sistem Monitoring Kesehatan Struktur (SMKS), Wireless Sensor Sensor Network (WSN), XBee Pro versi v1, Arduino Due, Ad hoc On demand Distance Vector (AODV), Destination Sequenced Distance Vector (DSDV) dan Network Simulator-2 (NS-2).

BAB 3. METODE PENELITIAN

Pada bab ini berisi metode penelitian yang digunakan dalam hal pengimplementasian manajemen konsumsi daya yang diperlukan. Komunikasi sensor node disimulaikan dengan NS-2 berdasarkan asumsi-asumsi yang dipaparkan pada bab ini serta menggunakan parameter dari datasheet XBee Pro versi 1.

BAB 4. HASIL DAN ANALISA

Pada bab ini berisi tentang hasil simulasi yang telah dilakukan beserta analisisnya. Simulasi yang dilakukan adalah berdasarkan desain yang telah dikemukakan pada bab 3 dengan menggunakan simulator NS-2 dan menggunakan parameter dari datasheet XBee Pro versi v1.

BAB 5. KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan penelitian dari analisa bab 3 beserta saran untuk kemajuan penelitian ini ke depannya

BAB 2 TINJAUAN PUSTAKA

2.1. Sistem Monitoring Kesehatan Struktur (SMKS)

Suatu bangunan memiliki masa ketahanan dalam beroperasi. Namun tidak menutup kemungkinan adanya penurunan atas kemampuan ketahanan yang dimiliki oleh bangunan tersebut, meskipun bangunan tersebut didesain agar dapat beroperasi untuk jangka waktu yang cukup lama. Tidak menutup kemungkinan penurunan performa mengakibatkan kerusakan pada badan bangunan, sehingga memerlukan biaya perbaikan yang tidak sedikit.

Berkurangnya kemampuan fisik dinilai dari perhitungan terhadap kondisi fisik kesehatan bangunan tersebut. Penilaian ini perlu dilakukan secara terus menerus agar dapat dilakukan suatu tindakan untuk mencegah terjadinya suatu hal yang tidak diinginkan. Maka dari itu perlu adanya suatu sistem untuk pemantauan kesehatan bangunan tersebut.

Seiring semakin majunya teknologi dalam bidang instrumentasi dan juga dalam bidang teknologi informasi dan komunikasi, maka sistem monitoring ini dapat dijalankan dengan lebih mudah. Terdapat suatu bidang baru dalam hal pemantauan kesehatan struktur dari kerusakan yakni Sistem Monitoring Kesehatan Struktur (SMKS).

2.1.1. SMKS Jembatan

Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan merupakan bidang pemantauan pada jembatan, dimana teknologi ini dapat mempepanjang umur pengoperasian jembatan karena dapat mendeteksi lebih awal tentang adanya kerusakan sehingga meminimalisir terjadinya kerusakan yang lebih parah. Beberapa tujuan dari SMKS ini adalah sebagai berikut [13]:

1. Menjamin keamanan struktur
2. Memperoleh perencanaan pemeliharaan struktur yang rasional dan ekonomis
3. Mencapai pekerjaan pemeliharaan yang aman dan ekonomis
4. Mengidentifikasi penyebab respon yang tidak dapat di terima

SMKS digunakan untuk pemantauan kondisi fisik jangka pendek maupun jangka panjang. Dalam praktiknya SMKS ini menggunakan beberapa sensor yang diletakkan pada badan jembatan yang kemudian digunakan untuk memperoleh respon dari struktur jembatan tersebut. Selanjutnya respon dari sensor akan di analisis untuk memperoleh informasi tentang tingkat kerusakannya. Terdapat beberapa faktor yang berkaitan dalam hal pengembangan dan kemajuan teknologi SMKS ini, yakni [14]:

1. Kemajuan terbaru dalam teknologi penginderaan dengan kecepatan tinggi dan sirkuit elektronik murah, dan pengembangan validasi sinyal yang sangat efisien dan pengolahan metode.
2. Perkembangan yang sedang berlangsung dalam teknologi komunikasi, yaitu banyak penggunaan internet dan teknologi nirkabel.
3. Perkembangan transmisi data dan sistem pengumpulan, serta sistem pengarsipan dan pengambilan data.
4. Kemajuan dalam pengolahan data, termasuk model deteksi kerusakan dan algoritma kecerdasan buatan.

2.1.2. Komponen Sistem Monitoring

Pada sistem monitoring mengacu pada pemantauan terus menerus menggunakan sensor yang ada, baik sensor yang tertanam pada badan jembatan ataupun yang melekat pada bagian luarnya. Sistem monitoring terdiri dari enam komponen, diantaranya :

1. Data Akuisisi

Meliputi berbagai jenis sensor dan pengumpulan data yang diinginkan.

2. Komunikasi data

Meliputi transmisi data dari site ke storage dan analisis lokasi. Sebagai contoh pengolahan informasi data jarak jauh.

3. Pengolahan dan analisis data

Meliputi pembersihan data dari gangguan dan informasi tambahan.

4. Penyimpanan data yang diolah

Data yang di peroleh disimpan untuk keperluan tahap selanjutnya.

5. Analisa diagnosa

Meliputi analisa pendeteksian kerusakan dan konversi data baru ke dalam respon structural.

6. Pengambilan informasi sesuai yang diperlukan

Meliputi pengambilan informasi dari hasil analisa diagnosa yang telah dilakukan.

2.1.3. Syarat Monitoring Jembatan

Pada sistem monitoring untuk mendapatkan data respon struktur sepanjang periode pengukuran untuk memverifikasi parameter beban stokastik dan respon struktur yang akan dibandingkan dengan respon yang dihitung.

Beberapa data digunakan untuk melihat kebenaran dari struktur atau untuk memverifikasi kekurangan yang ada. Monitoring dalam jangka waktu yang pendek meliputi pembebanan pada struktur atau monitoring beban yang tidak diharapkan. Contoh-contoh monitoring, diantaranya [13]:

1. Respon Stokastik

Karakteristik gempa, angin dan beban lalu lintas dan respon struktur dapat diukur di lapangan untuk memverifikasi prediksi yang dibuat pada model numerik pada tahap desain.

2. Beban Internal

Selain pengukuran permanen, pengukuran intensif dapat diulangi berkali-kali menggunakan mobile sensor untuk memetakan perubahan dalam distribusi gaya pada cable-stayed, tiang fondasi, dll. Distribusi regangan dapat dimonitor pada periode yang panjang untuk mengukur perubahan distribusi tegangan.

3. Respon *Fatigu* /Kelelahan

Beban fatigue pada sambungan las, dek dan balok diukur dengan strain gauge atau sistem accelerometer.

4. Respon Deterministik

Perpindahan pada buffer hidraulik, dampers, siar muai (expansion joint) yang tergantung dari temperatur dan distribusi beban pada dek ortotropik dapat dimonitor oleh sensor temperatur dan sensor displacement, tiltmeters dan sistem GPS.

5. *Global Static Response*

Respon statik pada fondasi, rangkai (creep) dan penyusutan (shrinkage), distribusi regangan pada kabel utama dapat dimonitor oleh sensor khusus.

Pengukuran dapat dilakukan untuk mengkalkulasikan seperti temperatur/regangan rata-rata dan perbedaan temperatur/regangan pada jarak yang jauh.

2.1.4. Klasifikasi Monitoring

SMKS dapat diklasifikasikan menjadi beberapa macam, diantaranya adalah klasifikasi menurut tingkat kecanggihannya, jenis informasinya dan pengambilan keputusan algoritma yang diberikan. Menurut [14] sistem monitoring berdasarkan tingkat kecanggihan dan kemajuan dapat di klasifikasi sebagai berikut:

1. Tingkat I

Pada tingkat 1 ini sistem monitoring dapat mendeteksi kerusakan dalam struktur, akan tetapi tidak dapat memberikan informasi tentang sifat, lokasi ataupun keparahan kerusakan. Sehingga pada tingkat ini belum bisa diambil penilaian tentang kesehatan dari suatu struktur tersebut.

2. Tingkat II

Untuk tingkat II ini sedikit lebih baik dari sistem pada tingkat I, pada tingkat ini dapat mendeteksi adanya kerusakan dan juga memberikan informasi tentang lokasi kerusakan.

3. Tingkat III

Pada tingkat III ini dapat mendeteksi dan menentukan kerusakan dan dapat memberikan beberapa indikasi dari beratnya.

4. Tingkat IV

Tingkat IV merupakan tingkatan yang paling canggih karena mampu memberikan informasi rinci tentang keberadaan, lokasi dan tingkat keparahan kerusakan sehingga informasi yang di dapat digunakan untuk mengevaluasi keamanan dari sistem.



Gambar 2.1 Klasifikasi sistem Monitoring Berdasarkan Kemampuan Pendeteksian dan Pengambilan Keputusan [14]

2.1.5. Parameter Akuisisi Data Dalam Monitoring Jembatan

Dalam pemantauan terdapat beberapa parameter penting yang perlu dipertimbangkan. Berikut ini menjelaskan beberapa parameter penting yang biasanya diperlukan untuk kebanyakan pengujian dan monitoring jembatan [13]:

1. Jumlah Sensor

Dalam setiap aplikasi pengujian atau pemantauan, jumlah sensor akan tergantung pada ukuran dan kompleksitas dari struktur dan pada kompleksitas perilaku yang sedang dievaluasi. Hal ini tidak biasa untuk aplikasi pengujian dan monitoring skala besar dan bahkan beberapa jembatan ukuran sedang yang memanfaatkan 120 atau lebih sensor. Karakteristik kinerja perangkat keras akuisisi data sering kali tergantung pada jumlah saluran sensor sedang dipindai.

2. Tipe Sensor

Jenis-jenis sensor yang akan digunakan akan menentukan banyak karakteristik yang dibutuhkan dari perangkat keras akuisisi data. Sebagai contoh, sebuah pengukur regangan resistansi memiliki karakteristik persyaratan output spesifik dan pengkondisian sinyal. Akuisisi data biasanya akan bervariasi untuk setiap jenis sensor yang digunakan.

Selanjutnya, dalam sebagian besar aplikasi, lebih dari satu jenis sensor akan dibutuhkan. Hal ini penting untuk memastikan bahwa karakteristik perangkat keras akuisisi data yang sesuai persyaratan untuk setiap jenis sensor sehingga pengukuran yang paling akurat dan diandalkan dapat diperoleh. Hal ini sering membutuhkan perangkat keras akuisisi data yang

cukup fleksibel untuk mengakomodasi berbagai bentuk peng-kondisian sinyal. Dalam beberapa kasus, penggunaan lebih dari satu jenis sistem akuisisi data dapat menawarkan solusi yang paling fleksibel dan pembiayaan yang efektif.

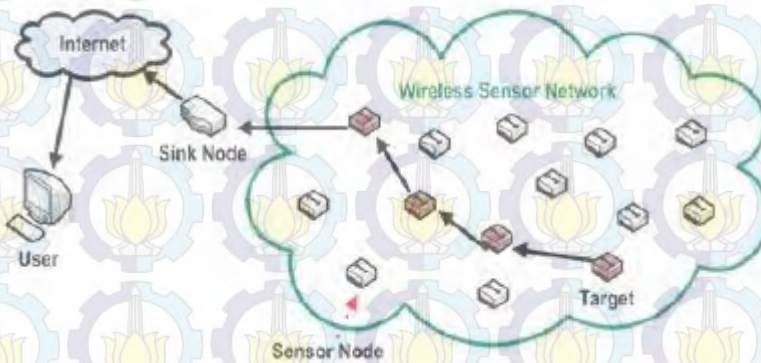
3. Mode Data Akuisisi

Ada dua tipe dasar mode akuisisi data, polled dan event-based. Pada mode polled, sistem akuisisi data umumnya dioperasikan secara terus menerus dan sinyal sensor yang sesuai dikumpulkan dan dicatat secara waktu berkala. Sensor yang mengukur variabel fisik yang statis atau yang perlahan-lahan bervariasi dengan waktu yang sesuai untuk mode polled akuisisi data. Sedangkan pada akuisisi data event-based, sinyal sensor dikumpulkan hanya selama terjadinya beberapa peristiwa penting yang ditetapkan pengguna. Peristiwa penting ini dapat acak atau deterministik dan beberapa contoh umum dari peristiwa tersebut meliputi kegiatan konstruksi, kontrol tes, sebuah truk berat melintasi jembatan atau suatu periode dengan volume lalu lintas berat, kecelakaan, dan kondisi lingkungan normal atau bencana alam seperti banjir, gempa bumi atau badai. Variabel fisik yang diukur oleh sensor selama terjadinya peristiwa seperti itu dapat berubah lambat atau cepat seiring waktu, tergantung pada kecepatan peristiwa.

2.2. *Wireless Sensor Network* (WSN)

Wireless Sensor Network (WSN) atau Jaringan Sensor Nirkabel merupakan suatu jaringan yang terdiri dari beberapa sensor yang masing-masing sensor tersebut memiliki kemampuan untuk merasakan (sensing), memproses serta berkomunikasi. Pada WSN, node sensor disebar dengan tujuan untuk menangkap adanya gejala atau fenomena yang hendak diteliti. Jumlah node yang disebar dapat ditentukan sesuai kebutuhan dan tergantung beberapa faktor misalnya luas area, kemampuan sensing} node, dan sebagainya. Tiap node memiliki kemampuan untuk mengumpulkan data dan meroutingkannya kembali ke Base Station}. Node sensor dapat mengumpulkan data dalam jumlah yang

besar dari gejala yang timbul dari lingkungan sekitar. untuk arsitektur WSN secara umum dapat direpresentasikan sebagai berikut:



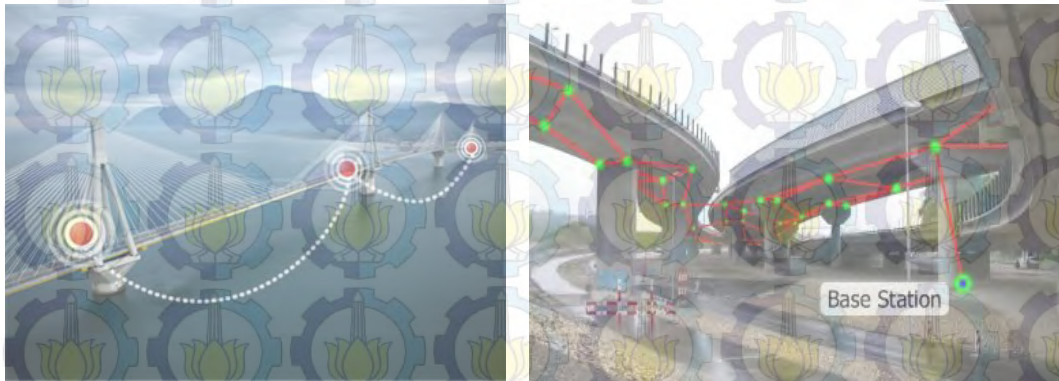
Gambar 2.2. Arsitektur WSN secara umum [15]

Pada Gambar 2.2, dapat dilihat, node sensor yang berukuran kecil disebar dalam di suatu area sensor. Node sensor tersebut memiliki kemampuan untuk merutekan data yang dikumpulkan ke node lain yang berdekatan. Data dikirimkan melalui transmisi radio akan diteruskan menuju BS (*Base Station*) yang merupakan penghubung antara node sensor dan user. Informasi tersebut dapat diakses melalui berbagai *platform* seperti koneksi internet atau satelit sehingga memungkinkan user untuk dapat mengakses secara *real time* melalui remote server.

Secara umum jaringan sensor nirkabel itu sendiri terdiri dari dua komponen, yaitu node sensor dan *sink*. Node sensor merupakan komponen kesatuan dari jaringan yang dapat menghasilkan informasi, biasanya merupakan sebuah sensor atau juga dapat berupa sebuah aktuator yang menghasilkan *feedback* pada keseluruhan operasi. Sink merupakan kesatuan yang mengumpulkan informasi dari node sensor sehingga dapat dilakukan pengolahan informasi lebih lanjut. Terdapat tiga bentuk sink yaitu *sink* dapat berupa node sensor yang lain dalam bentuk sensor dari jaringan itu sendiri atau dari jaringan lain. *Sink* dapat berupa sebuah komputer dan sebuah PDA yang digunakan untuk berinteraksi dengan Jaringan Sensor. Bahkan *sink* dapat berupa gateway ke jaringan yang lebih besar seperti internet sehingga interaksi dapat dilakukan melalui jarak yang sangat jauh dan tidak terkoneksi secara langsung dengan

Jaringan Sensor. Tugas utama dari sensor adalah memonitoring kondisi fisik dari suatu lingkungan dan menyampaikan hasilnya ke pusat kendali. Untuk data yang diperoleh bisa berupa pergerakan, tekanan atau temperatur.

Salah satu contoh WSN secara riil adalah digunakan pada *smart bridge* atau sistem SHM. Pada jembatan terdapat beberapa sensor node yang disebarakan pada beberapa titik jembatan seperti pada Gambar 2.3 (a) dan (b). Sensor node disini bertindak sebagai pendeteksi getaran dan temperatur. Jika terjadi masalah pada jembatan, seperti kabel yang terputus atau besi yang hilang, maka sebelum terjadi kecelakaan sensor node akan mengirimkan peringatan ke sink node atau base station seperti pada Gambar 2.3 (b).



a) WSN pada Long Span Bridge [24]

b) WSN pada Road Bridge [25]

Gambar 2.3. Contoh Aplikasi WSN

2.2.1. Karakteristik WSN

Pada WSN terdapat beberapa karakteristik yang diperlukan, diantaranya :

1. Jenis layanan *Type of Service*

Jenis layanan yang diberikan oleh jaringan komunikasi konvensional adalah bit bergerak dari satu tempat ke tempat lain.

2. *Quality of Service*

Erat kaitannya dengan jenis layanan jaringan adalah kualitas layanan tersebut. Kualitas layanan berhubungan dengan waktu pengiriman, yakni memperhatikan terhadap delay saat proses komunikasi data.

3. Skalabilitas

Skala WSN bervariasi, maka arsitektur dan protokol harus mampu bekerja dari skala kecil sampai skala besar.

4. Programmability

Node-node sensor dapat di program sesuai dengan keadaan lingkungan yang akan diamati.

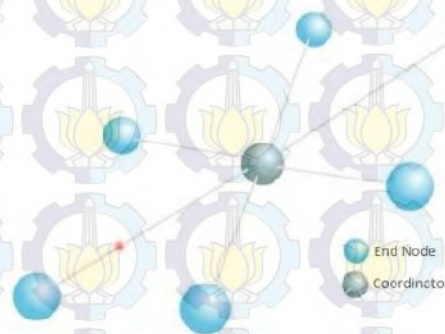
2.2.2. Topologi Jaringan WSN

Topologi jaringan adalah suatu cara untuk menghubungkan titik satu dengan titik lainnya. Masing-masing topologi mempunyai ciri khas, kelebihan dan kekurangan. Topologi yang di gunakan pada jaringan sensor nirkabel diantaranya:

1) Topologi *Star*

Topologi star adalah topologi paling dasar. Topologi star terdiri dari koordinator dan beberapa perangkat akhir (node). Dalam topologi ini, perangkat akhir berkomunikasi hanya dengan koordinator. Setiap pertukaran paket antara perangkat akhir harus melalui koordinator.

- Kerugian dari topologi ini adalah pengoperasian jaringan tergantung pada koordinator jaringan, dan karena semua paket antara perangkat harus melalui koordinator, koordinator mungkin menjadi bottlenecked. Juga, tidak ada jalan alternatif dari sumber ke tujuan.
- Keuntungan dari topologi star adalah paket melalui paling banyak dua hop untuk mencapai tujuan mereka.

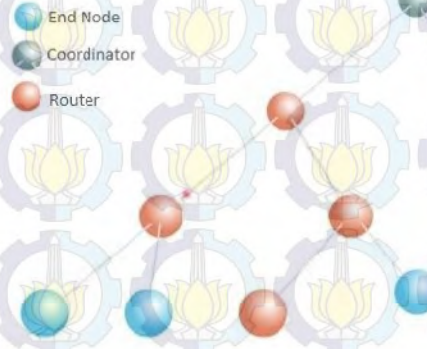


Gambar 2.4 Topologi Star [16]

2) Topologi *Tree*

Topologi tree lebih kompleks dibandingkan dengan topologi star. Setiap node masih mempertahankan satu jalur komunikasi untuk *gateway*, perbedaannya menggunakan node-node lain dalam mengirimkan data, namun masih dalam satu jalur tersebut.

- Kelemahan untuk topologi ini adalah jika node router yang down, maka semua node yang bergantung pada node router akan kehilangan komunikasi ke gateway.

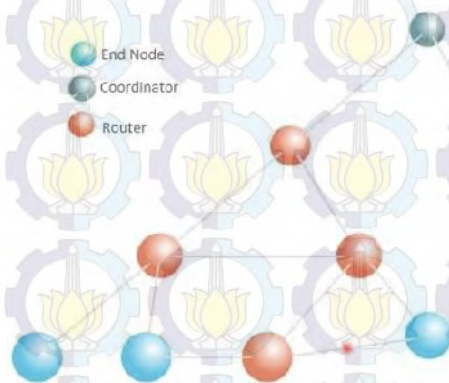


Gambar 2.5 Topologi Tree [16]

3) Topologi *Mesh*

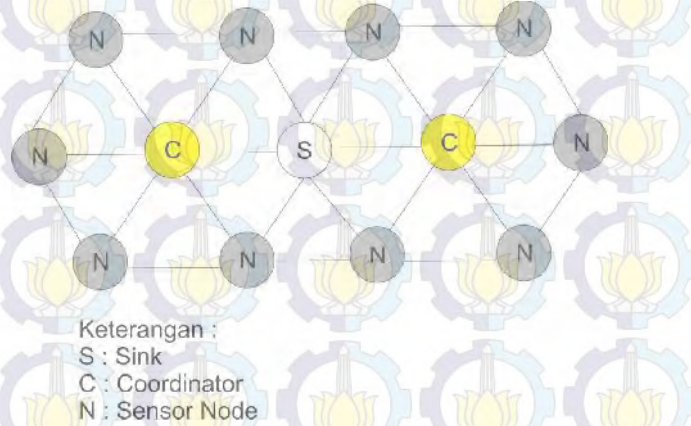
Topologi *Mesh* juga disebut sebagai jaringan peer-to-peer, terdiri dari satu koordinator, beberapa router, dan perangkat akhir. Topologi ini merupakan jalur komunikasi dimana masing-masing node dapat berkomunikasi dengan yang lainnya.

- Kelebihan dari topologi ini adalah dapat meningkatkan kehandalan sistem. Dalam sebuah jaringan *mesh*, node mempertahankan jalur komunikasi untuk kembali ke gateway, sehingga jika salah satu node router *down*, secara otomatis router data akan dilewatkan melalui jalur yang berbeda.
- Kelemahan pada topologi ini adalah adanya latensi atau delay karena data harus melalui beberapa hop sebelum mencapai *gateway*.

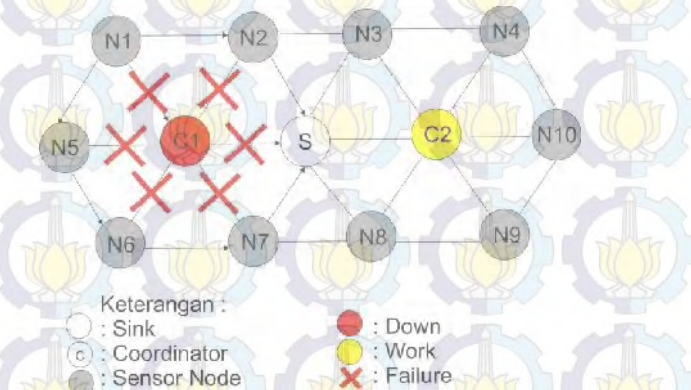


Gambar 2.6 Topologi *Mesh* [16]

Contoh lain dari topologi *mesh* dapat di lihat pada Gambar 2.7, dimana jaringan terdiri dari satu sink, dua *coordinator* atau *router* dan 10 sensor node. Antar node saling terhubung dan juga terhubung pula dengan *coordinator* dan sink.



Gambar 2.7. Topologi *Mesh* dengan 2 *coordinator* dan *sink*



Gambar 2.8. Topologi *Mesh* terdapat *coordinator* yang *down*

Melihat dari kelebihan topologi *mesh* yakni pada pernyataan bahwa topologi *mesh* mampu mempertahankan jalur komunikasi untuk kembali ke gateway, sehingga jika salah satu node router *down*, secara otomatis router data akan dilewatkan melalui jalur yang berbeda maka Seperti pada Gambar 2.8, ketika sensor node (N1) ingin mengirimkan data menuju sink melalui *coordinator* / *router* (C1). Namun, karena *coordinator* (C1) *down*, maka data tetap dapat dikirimkan menuju sink. Dengan topologi *mesh*, data dari N1 dapat dikirimkan menuju sink melalui N2 atau N5. Jika data diteruskan pada N2, maka data akan langsung dikirimkan menuju sink. Namun, jika data

diteruskan menuju N5, maka data akan diteruskan lagi ke N6, N7 dan terakhir menuju sink. Kelemahan pada topologi ini adalah adanya keterlambatan / delay karena data harus melalui beberapa hop sebelum mencapai *gateway* dimana pada Gambar 2.10 terdapat forward data dari N5, N6, N7, baru mencapai sink.

2.3. DSDV

DSDV merupakan kependekan dari *Destination Sequenced Distance-Vector* adalah algoritma routing protocol adhoc proaktif yang didasari pada Bellman – Ford yang pertama kali dikenalkan. Algoritma ini berkontribusi untuk mengatasi *Routing Loop*. Pada DSDV, digunakan *sequence number* untuk mengirimkan pesan pada jaringan. *Sequence number* dihasilkan juga saat ada perubahan dalam jaringan, hal ini terjadi karena sifat table routing node pada jaringan yang menggunakan protokol proaktif yang update secara periodik, serta *Trigered update* ulang digunakan oleh node untuk mengupdate node yang masuk dan keluar dari jaringan [10].

DSDV merupakan salah satu *Proactive Routing Protocol* yang memerlukan setiap node untuk mengirimkan tabel routing ke seluruh node tetangganya secara periodik. Metode routing DSDV yakni node yang berada dalam jaringan akan menjaga tabel routing ke node tetangganya. Tabel routing berisi alamat node tujuan, jumlah hop yang diperlukan agar sampai di tujuan dan *sequence number*. Saat tabel routing dalam satu node telah diupdate, maka node akan memilih rute untuk mencapai node tujuan dengan beberapa kriteria :

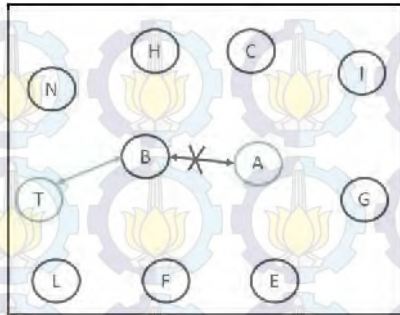
- Memiliki *sequence number* terbaru dengan melihat *sequence number* yang paling besar.
- Apabila nilai *sequence number* sama, maka akan dilihat nilai metricnya. Nilai *metric* yang dipilih adalah yang lebih kecil.

DSDV memiliki beberapa kelebihan, diantaranya [10]:

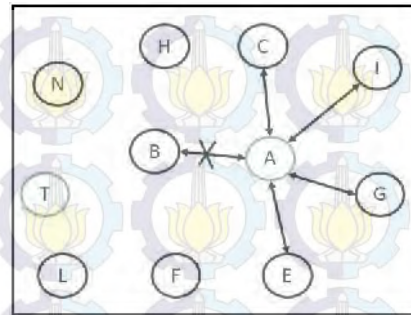
- DSDV menjamin tidak ada *looping route*
- DSDV dapat mereduksi masalah *count to infinity*
- DSDV dapat menghindari trafik lebih dengan kenaikan drastis update penuh untuk dump

- DSDV hanya maintenance path terbaik menuju tujuan, dari sekian banyak path ketujuan.

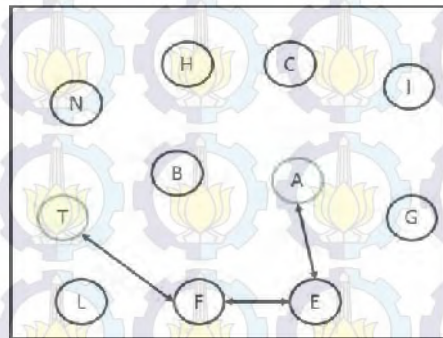
Ilustrasi pengiriman paket dengan routing protokol DSDV yakni ketika akan menyampaikan pesan dari node "A" untuk ke tujuan node "T" namun terdapat jalur yang breaks, maka protokol membuat link sementara melalui node tetangga yang memiliki rute valid menuju tujuan yang dikehendaki. Link sementara dibuat dengan mengirimkan satu-hop ROUTE-REQUEST dan pesan ROUTE-ACK. Node "A" ketika menemukan link hop yang rusak maka node akan menyebarkan (broadcast) satu hop paket ROUTE-REQUEST ke semua node tetangganya. Selanjutnya, node tetangga mengembalikan ROUTE-ACK didapatkan informasi bahwa terdapat rute untuk sampai ke tujuan namun node "A" bukan termasuk dalam rute tersebut. Setiap masukan pada tabel routing mempunyai masukan tambahan untuk waktu update rute. Waktu ini berada pada ROUTE-ACK yang digunakan dalam memilih rute sementara. Pada beberapa kasus terkadang ROUTE-ACK menerima update rute dengan nomor urut (sequence number) yang sama dan hop minimum, mengatasi hal ini maka node akan memilih rute dengan update rute terbaru. Pada Gambar 2.9 menunjukkan bagaimana node "A" membuat rute sementara untuk menuju node "T" ketika link diantaranya yakni dari node "A" ke node "B" breaks. Disini node "A" menunda terlebih dahulu pengiriman pakatnya sebagaimana ditunjukkan pada Gambar 2.9a. Selanjutnya node "A" menyebarkan (broadcast) paket ROUTE-REQUEST kepada node tetangganya. Node C, E, dan G merespon dengan mengirimkan paket ROUTE-ACK kembali waktu update rute dan hop ke node "A" seperti di tunjukkan pada Gambar 2.9b. Tabel 2.1 menunjukkan informasi update rute yang di terima oleh node "A". Dari data yang diperoleh pada tabel dapat dilihat bahwa node "C" dan "E" memiliki nilai yang sama untuk jumlah hop yang dapat dilewati, akan tetapi waktu update rute pada node "E" lebih besar dibandingkan node "C", yang berarti rute melalui E adalah update rute terbaru. Sehingga untuk pengiriman paket dari node "A" menuju node "T", node "A" memilih node "E" sebagai node selanjutnya seperti terlihat pada Gambar 2.9c.



a) Link Dari Node A ke B Breaks



b) Node A Menyebarkan ROUTE-REQUEST ke Node Tetangga



c) Node A Memilih Node E Sebagai Hop Selanjutnya

Gambar 2.9 Pembuatan Rute Sementara Node A

Tabel 2.1 Update Rute pada Node A

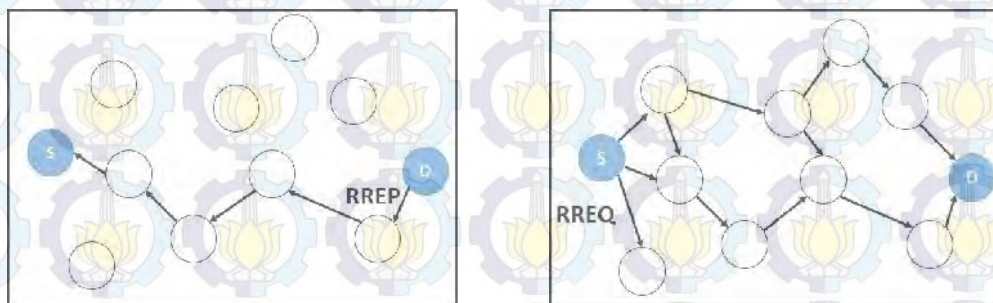
Node Tetangga	Jumlah Hop	Node Selanjutnya	Waktu Update (ms)
C	2	H	1765
E	2	F	1860
G	3	E	1050

2.4. AODV

Ad hoc On-demand Distance Vector (AODV). AODV adalah distance vector routing protocol yang termasuk dalam klasifikasi reaktif routing protocol, yang hanya me-request sebuah rute saat dibutuhkan. AODV yang standar ini dikembangkan oleh C. E. Perkins, E.M. Belding-Royer dan S. Das pada RFC 3561 [12].

Karena AODV merupakan on-demand, sebuah rute dibangun hanya jika dibutuhkan oleh node sumber untuk mentransmisikan paket data dan AODV menjaganya selama rute ini dibutuhkan. AODV menggunakan *sequence number* yang dibuat node tujuan untuk menentukan jalur terbaru ke node tujuan. Sebuah node akan memperbarui informasi rute hanya jika *sequence number* tujuan paket yang diterima sekarang lebih besar dibandingkan *sequence number* yang disimpan pada node. Hal ini mengindikasikan barunya rute yang diterima oleh node sumber. Untuk menghindari pengiriman ganda pada paket yang sama, AODV menggunakan nomor identitas pengiriman yang menjamin bebas *looping* karena node intermediate hanya meneruskan salinan pertama dari paket yang sama dan membuang duplikasi salinan.

Untuk menentukan rute ke node tujuan, node sumber mengirimkan paket *Route Request* (RREQ) ke jaringan. RREQ berisi identitas node sumber, identitas node tujuan, nomor urut tujuan, nomor urut sumber, identitas pengiriman dan TTL (*Time to Live*). Node yang menerima RREQ baik itu node tujuan atau tidak, node tersebut memiliki rute terbaru ke node tujuan dapat merespon ke RREQ untuk mengirimkan paket *Route Reply* (RREP) kembali ke node sumber.



a) Node sumber S mengirim RREQ ke node tetangganya

b) Node tujuan mengirimkan RREP kembali ke sumber

Gambar 2.10 Mekanisme Penemuan Rute

Ketika sebuah node meneruskan paket RREQ ke node tetangganya, node tersebut juga mencatatnya pada tabel routing dari node mana salinan RREQ itu datang. Informasi ini digunakan untuk membangun reverse path untuk paket RREP. AODV hanya menggunakan link yang simetris karena paket RREP mengikuti reverse path dari paket RREQ. Ketika sebuah node menerima sebuah paket RREP, informasi tentang node sebelumnya darimana paket RREP tersebut

di terima juga disimpan dengan tujuan untuk meneruskan paket data ke node berikutnya sebagai hop berikutnya menuju tujuan. ketika node sumber menerima sebuah paket RREP, rute tersebut sudah siap digunakan untuk mengirimkan data.

Selanjutnya node sumber mengirimkan kembali paket RREQ jika tidak menerima paket RREP sebelum kadaluarsa. Node sumber melakukan pencarian rute dengan usaha yang maksimal. Jika node sumber berpindah maka node sumber bisa memulai kembali pencarian rute node tujuan [9]. Mekanisme penemuan rute ditunjukkan pada Gambar 2.10.

2.5. Zigbee (802.15.4)

Zigbee adalah salah satu protokol pada jaringan nirkabel yang didesain oleh *zigbee* Alliance. *Zigbee* juga merupakan protokol yang telah dirancang khusus untuk komunikasi jarak pendek di WSN. Semua lapisan pada *zigbee* didasarkan pada standar IEEE 802.15.4. Lapisan pada *zigbee* meliputi lapisan fisik, lapisan jaringan, lapisan aplikasi dan lapisan keamanan. *Zigbee* memiliki keunggulan pada bentuknya yang minimalis dan pengoperasiannya yang mudah. *Zigbee* didesain untuk melakukan komunikasi jarak pendek, yaitu dengan jarak komunikasi hanya sekitar 50 meter hingga 100 meter. Sedangkan kecepatan komunikasi yang dapat dilakukan *zigbee* hanyalah 250 kbps. Terdapat perbedaan jika kecepatan *zigbee* dibandingkan dengan sistem komunikasi jarak pendek lainnya, misalnya Wi-Fi yang memiliki kecepatan komunikasi hingga 54 Mbps.

2.5.1. Karakteristik ZigBee

ZigBee memiliki karakteristik diantaranya :

- Bekerja pada tiga rentang frekuensi yakni frekuensi 2.4 GHz, 868MHz dan 915MHz. Frekuensi 868-870 MHz dengan 1 kanal, frekuensi 902-928 MHz dengan 10 kanal dan frekuensi 2,4 GHz dengan 16 kanal (digunakan di Indonesia).
- Mempunyai konsumsi daya yang rendah
- Maksimum transfer rate untuk tiap data pada tiap lebar pita adalah sebagai berikut 250Kbps untuk 2.4GHz, 40 kbps untuk 915 MHz, dan 20Kbps untuk 868 MHz

- Mempunyai Throughput yang tinggi dan dan latency yang rendah untuk duty cycle yang kecil.



Gambar 2.11 Perangkat Zigbee

2.5.2. Arsitektur ZigBee

Arsitektur lapisan komunikasi pada zigbee berdasarkan standar IEEE 802.15.4 terdapat beberapa lapisan yang terdiri dari lapisan berikut ini :

1. Physical Layer

Physical Layer (Layer Fisik) merupakan komponen yang sangat penting dalam proses komunikasi antar perangkat. Layer ini biasanya digunakan untuk mengubah data logika menjadi bentuk yang sesuai untuk dikirimkan pada media transmisi yang digunakan. Juga sebagai interface (antarmuka) dan penentu kualitas dari sebuah komunikasi, sekaligus bertugas mensuplai berbagai macam informasi dari layer di atasnya.

2. Media Access Control (MAC) Layer

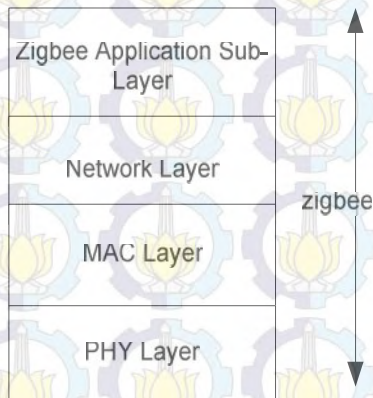
Layer ini digunakan untuk mengakses saluran yang digunakan. Dimana terdapat dua mekanisme untuk mengakses saluran, yaitu mode Beacon yang menggunakan teknik CSMA/CA dan mode non-Beacon yang menggunakan teknik non CSMA/CA.

3. Network Layer.

Network Layer digunakan untuk mengatur jaringan, antara lain : konfigurasi perangkat, pengalamatan, penggabungan jaringan, sistem keamanan jaringan.

4. Application Layer

Application Layer digunakan untuk mencocokkan antar dua perangkat yang berkomunikasi dalam waktu bersamaan dan menyampaikan pesan antar dua perangkat tersebut.



Gambar 2.12. Arsitektur ZigBee

2.5.2. XBee Pro versi v1

XBee Pro versi v1 merupakan salah satu produk dari Digi International, Inc. yang mendukung dan memenuhi standar IEEE 802.15.4 dan dirancang untuk komunikasi tanpa kabel dengan band frekuensi 2,4 GHz. Salah satu keunggulan XBee Pro ini adalah konsumsi daya yang sangat rendah, sehingga memiliki lifetime yang sangat lama. Pada XBee Pro versi v1 memiliki spesifikasi sebagai berikut :

XBee Pro versi v1 digunakan dalam komunikasi data antar sensor node maupun komunikasi data antara sensor node dengan sink. Pada *transceiver* XBee modul mempunyai dua mode operasi :

- 1) Transparent serial port mode. Pengiriman data dari sensor ke modul XBee melalui serial port, kemudian XBee module mengirimkan data ke module XBee lainnya secara *wireless*.
- 2) Packet mode. Pengiriman pesan ke module XBee itu sendiri. Terdapat dua macam *packet* mode, yaitu IO packet dan command packet.

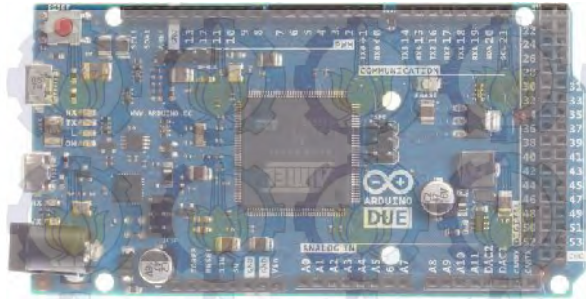
Tabel 2.2 Spesifikasi Modul XBee Pro versi v1 [13]

No	Spesifikasi	Nilai
1)	Transmit Power Output	<ul style="list-style-type: none">• 63 mW (18 dBm) pada wilayah Eropa• 10 mW (10 dBm) untuk international
2)	Receiver Sensitivity	-100 dBm
3)	RF Data Rate	250.000 bps
4)	Supply Voltage	2,8 – 3,4 V
5)	Transmit Current	250 mA (@3,3 V)
6)	Receive Current	55 mA (@3,3 V)
7)	Generating Frequency	ISM 2,4 GHz
8)	Antenna Type	Omni-directional Gain : 2,1 dBi

RF Modul Operation. Komunikasi XBee/ XBee-Pro ke host device adalah melalui sebuah logic-level asynchronous serial port. Melalui serial port ini, modul dapat berkomunikasi dengan beberapa logic atau voltage compatible UART, atau melalui sebuah level translator ke beberapa perangkat serial (misalnya: melalui RS 232 atau USB interface board).

2.6. Arduino Due

Arduino Due diperkenalkan pada tahun 2005 dan dirancang dengan harga yang murah dan didesain untuk penggunaan sensor dan aktuator. Arduino Due merupakan board mikrokontroler yang berbasis CPU Atmel SAM3X8E ARM Cortex - M3. Arduino due ini merupakan board pertama dengan 32-bit ARM core microcontroller yang memiliki 54 pin input/output digital (12 pin digunakan untuk output PWM), 12 input analog, 4 UART (serial port hardware), 84 MHz clock, koneksi USB OTG, 2 DAC (Digital to Analog), 2 TWI, power jack, SPI header, JTAG header, dan tombol *reset* dan *delete* [26]. Sedangkan Tabel 2.2 merupakan ringkasan dari spesifikasi Arduino Due.



Gambar 2.13. Arduino Due

Tabel 2.3. Spesifikasi Arduino Due

No.	Parameter	Nilai
1)	Mikrokontroler	AT91SAM3X8E
2)	Operating Voltage	3,3 V
3)	Input Voltage (recommended)	7 – 12 V
4)	Input Voltage (limits)	6 – 16 V
5)	Digital I/O Pins	54 (of which 12 provide PWM output)
6)	Analog Input Pins	12
7)	Analog Output Pins	2 (DAC)
8)	Total DC Output Current on all I/O lines	130 mA
9)	DC Current for 3,3 V Pins	800 mA
10)	DC Current for 5 V Pins	800 mA
11)	Flash Memory	512 KB all available for the user applications
12)	SRAM	96 KB (two banks : 64 KB and 32KB)
13)	Clock Speed	84 Hz

2.7. Sensor

Sensor secara umum merupakan alat untuk mendeteksi atau mengukur besaran fisis seperti panas, sinar, kimia, getaran dan lain-lain dan mengubahnya menjadi sinyal elektrik seperti arus listrik maupun tegangan. Sensor pada SMKS

yang umumnya diimplementasikan terdapat beberapa macam diantaranya *accelerometer, strain gauges, temperature sensor, displacement transducers, level sensing station, anemometer, dynamic weight-inmotion sensor*. Namun, pada penelitian ini hanya menfokuskan pada sensor accelerometer saja.

2.7.1 Sensor Accelerometer

Sensor accelerometer merupakan sensor yang digunakan untuk mengukur percepatan, mendeteksi dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi (inklinasi). Sensor accelerometer dibutuhkan untuk mengukur getaran yang terjadi pada jembatan dalam implementasi sistem SHM jembatan.

Biasanya, jembatan mengalami getaran saat terdapat beban atau kendaraan yang melewati jembatan. Pada [26], sensor accelerometer yang digunakan dengan tipe MMA 7361 dengan triple axis dimana accelerometer akan mendeteksi getaran dengan arah sumbu-X, sumbu-Y dan sumbu-Z.

2.8. Standar IEEE 802.15.4

Standar IEEE 802.15.4 bertujuan menghasilkan harga yang murah dan energi komunikasi yang kecil. Pada standar IEEE 802.15.4 tahun 2003, ditetapkan dua pilihan *physical layer* di pita frekuensi berbeda namun MAC bekerja efektif.

Pada tahun 2006, terdapat tambahan beberapa pilihan *physical layer*, sedangkan frame pada MAC mengalami peningkatan versi, peningkatan keamanan, dan didukung basis *shared-time* dengan metode *stamping time*, penjadwalan beacon dan sinkronisasi pesan *broadcast beacon* pada *beacon enabled* [25].

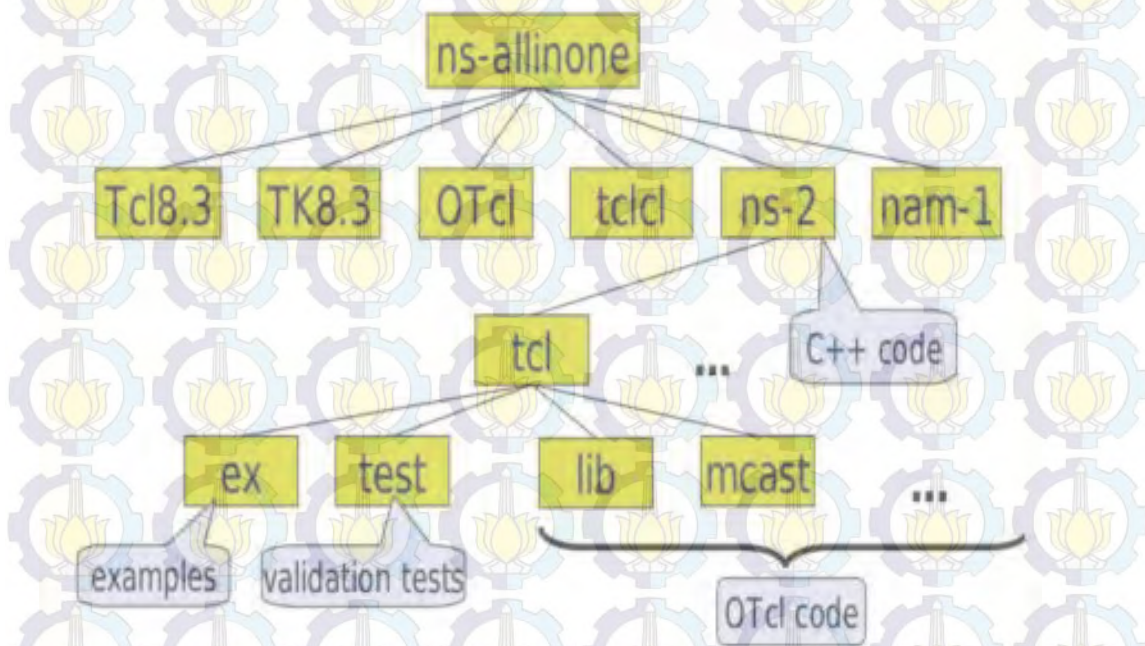
2.9. Network Simulator-2 (NS-2)

Network Simulator (NS) pertama kali dibangun sebagai varian dari REAL Network Simulator pada tahun 1989 di UCB (University of California Berkeley). NS2 merupakan suatu sistem yang bekerja pada OS Unix/Linux namun juga bisa dijalankan pada OS Windows namun harus menggunakan Cygwin Linux *Enviromentnya*. NS2 dibangun dari 2 bahasa pemrograman yaitu C++ dan OTcl, C++ sebagai *library* yang berisi *event scheduler*, protokol, dan komponen

jaringan yang diimplementasikan pada simulasi oleh user . Sedangkan OTcl digunakan pada script simulasi yang ditulis oleh NS user .Otcl juga berperan sebagai interpreter. Bahasa C++ digunakan pada library karena C++ mampu mendukung runtime simulasi yang cepat, meskipun simulasi melibatkan simulasi jumlah paket dan sumber data dalam jumlah besar. Sedangkan bahasa Tcl memberikan respon runtime yang lebih lambat daripada C++, namun jika terdapat kesalahan, respon Tcl terhadap kesalahan syntax dan perubahan script berlangsung dengan cepat dan interaktif.

2.9.1. Komponen Pembangun NS2

Untuk komponen pembangun NS2 dapat dilihat pada Gambar 2.11 berikut :



Gambar 2.14 Komponen Pembangun NS2

Keterangan:

- Tcl : Tool command language
- OTcl : Object Tcl
- TK : Tool Kit
- Tclcl : Tcl/C++ Interface
- NS2 : NS versi 2
- Nam : Network animator

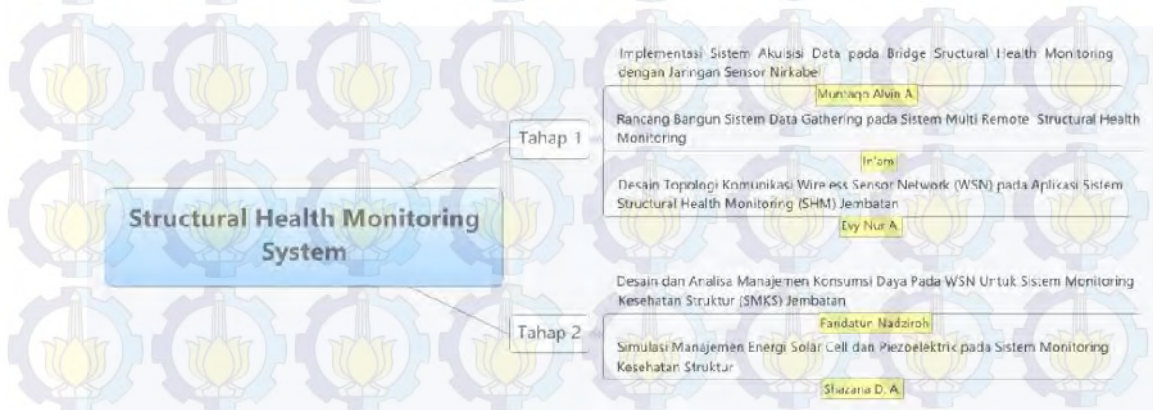
Pada Gambar 2.11 terlihat bahwa komponen pembangun NS2 yakni pada file folder ns-allinone terdapat beberapa *tool* yakni Tcl8.3 sebagai *tool command*, Tk8.3 sebagai *tool kit*, Otcl sebagai *object tcl*, Tctl sebagai *interface*, NS2 sebagai simulatornya dan nam sebagai GUI yang menampilkan hasil simulasi. Pada *tools* Tctl terdapat file tcl dan C++ yang berisi program atau kode-kode. Pada Tcl terdapat beberapa file di dalamnya yakni Ex yang berisi contoh-contoh program yang sudah ada, Test untuk validasi, Lib yang merupakan *library* dan Mcast. Lib dan Mcast termasuk pada pengkodean Otcl. Tcl adalah bahasa pemrograman yang didasarkan pada string-string based command. Tcl di desain untuk menjadi perekat dalam membangun software *building block* untuk menjadi suatu aplikasi. Sedangkan Otcl adalah ekstensi tambahan pada Tcl yang memungkinkan fungsi *object oriented*.



BAB 3 METODE PENELITIAN

3.1. Rancangan Penelitian

Sistem monitoring kesehatan struktur sudah diterapkan pada berbagai objek bangunan, dengan *constrain* yang berbeda menjadikan sistem dalam monitoring pun berbeda mengikuti objek yang diamati. Penelitian ini berfokus pada jembatan bentang panjang Suramadu yakni jembatan yang menghubungkan antara pulau Jawa dan Madura. dalam penelitian ada banyak hal yang perlu diperhatikan dari jembatan seperti kondisi lingkungan dan struktur. Penelitian ini merupakan penelitian besar. Beberapa penelitian yang telah dilakukan pada Jembatan Suramadu ini dapat dilihat pada road map pada Gambar 3.1 di bawah ini.

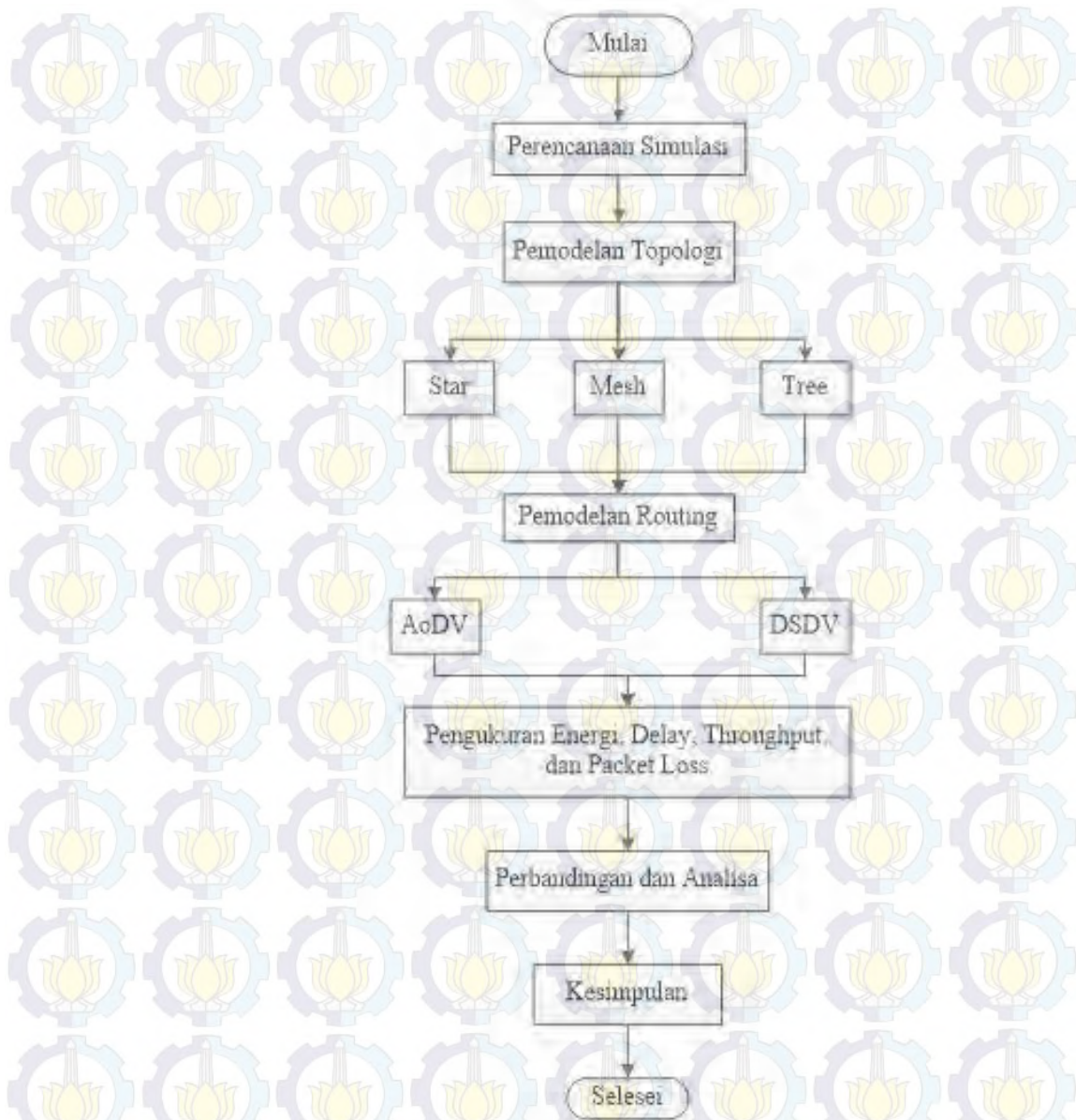


Gambar 3.1 Road Map Penelitian

Tahap awal penelitian terdapat beberapa fokus penelitian yang telah dilakukan. Pertama yakni melakukan penelitian tentang sistem akuisisi data pada sistem monitoring kesehatan struktur Jembatan Suramadu, dengan memastikan data sensor dapat diterima oleh server. Kedua yakni penelitian tentang pemantauan secara terpusat pengiriman data sensor ke server yang di fokuskan tidak hanya pada satu jembatan saja, melainkan beberapa jembatan dapat dipantau secara terpusat. Ketiga melakukan simulasi jaringan untuk mengetahui topologi dan desain routing terbaik dan cocok untuk diterapkan pada sistem monitoring kesehatan struktur jembatan[17]. Mengacu dari tahap awal penelitian, belum adanya pengaturan energi yang digunakan. Manajemen energi berperan penting

dalam jaringan sensor nirkabel, agar penggunaan energi lebih efisien maka perlu adanya teknik untuk manajemen energi yang digunakan. Karena menggunakan teknologi nirkabel tanpa adanya kabel yang terpasang ke jala-jala PLN membuat jaringan sensor nirkabel menggunakan baterai sebagai sumber daya. Keberadaan sumber daya ini memiliki *durability* yang terbatas dan perlu adanya pergantian. Maka pada tahap kedua dilakukan manajemen energi pada konsumsi daya dan sumber dayanya. Pada tahap kedua ini dilakukan penelitian untuk manajemen energi dengan cara mengatur daya yang dikonsumsi oleh jaringan sensor nirkabel seminimal mungkin menggunakan routing dan topologi yang tepat. Seperti ketika node sensor belum melebihi *threshold* maka berstatus *idle* sehingga daya yang dikonsumsi tidak sebanyak ketika pengiriman data ataupun ketika node sensor aktif. Namun pada penelitian tahap pertama pada bagian ketiga [17] masih terdapat kekurangan yakni untuk desain peletakan sensor jembatan yang digunakan belum memenuhi gambaran Jembatan Suramadu secara *real time*. Maka pada penelitian ini dilakukan desain ulang peletakan sensor dengan mengacu pada Jembatan Suramadu tersebut. Sebelum melakukan simulasi dari penelitian, dirancang terlebih dahulu alur penelitian yang akan dilakukan. Tahapan-tahapan yang akan dilakukan dapat dilihat pada Gambar 3.2.

Penelitian dimulai dengan perencanaan simulasi meliputi pemodelan sistem yang akan digunakan diantaranya jenis antrian, banyaknya paket data dan banyak node, selanjutnya pemodelan topologi memanfaatkan topologi star, tree dan mesh dengan menggunakan karakteristik radio module berupa Zigbee (802.15.4). Setelah pemodelan topologi, dilakukan pemodelan routing protokol. Terdapat dua macam routing protokol yang akan digunakan yakni AODV dan DSDV. Pada setiap routing protokol dilakukan pengukuran beberapa parameter yakni energi, *delay*, *throughput* dan *packet loss*. Setelah di dapat hasil dari parameter yang diukur, selanjutnya menganalisa hasil pengukuran dan kinerja dari routing protokol yang kemudian kedua hasil tersebut di bandingkan untuk di dapatkan kesimpulan.



Gambar 3.2 Flowchart Rencana Penelitian

Untuk proses manajemen energi dari node sensor, dilakukan dengan cara mengatur desain topologi dan routing yang digunakan.

3.2. Manajemen Sumber Daya

Sebuah sistem pada jaringan sensor nirkabel memiliki kemandirian untuk sumber daya yang dikonsumsi. Namun terdapat permasalahan penting yakni dalam proses bagaimana energi yang terbatas pada jaringan sensor nirkabel ini

dapat memberikan daya yang optimal. Pada penelitian [26] penghematan konsumsi daya dilakukan dengan cara mengurangi jumlah data pengiriman, penjadwalan pemancar radio dan komponen sensor, penambahan interval waktu pengiriman data dan deteksi sensor. Dengan menggunakan penelitian sebelumnya sebagai acuan dalam penentuan konsumsi daya yang digunakan oleh jaringan sensor nirkabel maka dapat diketahui berapa daya yang harus dicover. Penggunaan energi yang dikonsumsi digolongkan dalam beberapa macam, yakni :

1. Menghindari dan menghemat aktifitas yang membuang energi.
2. Mengestimasi penggunaan energi pada semua subsistem dan akibatnya pada *lifetime* jaringan sensor nirkabel.

Sedangkan konsumsi daya yang diperlukan oleh sistem, yakni :

Tabel 3.1 Konsumsi Daya pada Sistem

Karakteristik	Spesifikasi	Daya (Volt)
Sensing	ADXL345	2 - 3.6
	Strain gage	1
Komputing	Arduino due SAMX8E	3.3 - 5
Komunikasi	Xbee pro	2.8 - 3.4
Total Daya		13 Volt

Dengan mengacu pada Tabel 3.1, maka pada penelitian ini dilakukan metode untuk penghematan energi dengan menggunakan metode pengaturan routing dan topologi yang sesuai dengan desain jembatan. Teknik manajemen konsumsi daya didapatkan melalui pengaturan node-node sensor dengan memanfaatkan kondisi *sleep/awake* pada node sensor tersebut. Node *awake* ketika terjadi proses pentransmisian data dengan node tetangganya. Sedangkan untuk node yang tidak mentransmisikan data dalam kondisi *sleep*.

3.3. Metode Pengiriman Data

Pada proses pengiriman data perlu diperhatikan konsumsi daya yang diperlukan oleh node. Selain proses manajemen energi sumber daya, perlu adanya suatu metode untuk memajemen konsumsi dayanya. Suatu node apabila berada pada kondisi selalu aktif maka dapat menghabiskan *supply* daya yang ada. Sehingga perlu adanya suatu metode untuk menghemat konsumsi daya tersebut. Pada penelitian ini ditawarkan dengan model komunikasi berdasarkan terjadinya

suatu kejadian. Node yang telah dibagi dalam 2 macam yakni node sensor dan node koordinator/*sink*. Node sensor memiliki fungsi sebagai sensing node yang akan mensensing adanya getaran yang tertangkap di daerah sensing node. Node sensor yang berada paling ujung (*End device*) melanjutkan pengiriman datanya melalui node sensor terdekatnya. Node sensor yang berada dekat dengan node koordinator (router) akan menerima data dari node sensor sebelumnya dan data hasil pembacaan sensor dikirimkan pada node koordinator. Sedangkan node koordinator berfungsi sebagai pengumpul data sensing dari node sensor yang kemudian data akan dikirimkan pada base station. Berikut metode pengiriman data pada WSN :

1. Node sensor berada pada kondisi *stand by* untuk melakukan proses penginderaan getaran.
2. Ketika ada mobil melintas dan mulai memasuki area node sensor dan berada pada posisi node sensor maka node sensor mulai melakukan penginderaan terhadap getaran.
3. Setelah keluar dari area node maka data akan dikirim ke node koordinator.
4. Kemudian data di teruskan ke base station.

Disini terdapat banyak node sensor yang diletakkan berseberangan, ketika node sensor sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *awake* (node aktif), sedangkan ketika node sensor tidak sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *sleep*. *Sleep* bukan berarti node mati namun node berada pada kondisi *idle* atau hanya mendengar saja. Metode ini mampu menghemat daya yang dibutuhkan oleh node. Karena pada kondisi *awake* dan *sleep* lebih banyak membutuhkan daya pada kondisi *awake*.

3.4. Pemodelan Sistem

Pada penelitian ini dilakukan simulasi dengan menggunakan sistem WSN dengan membuat node sederhana menggunakan parameter dibawah ini:

Jenis kanal	: kanal Wireless
Jenis propagasi	: propagasi TwoRayGround
Jenis layer fisik	: 802.15.4
Jenis layer MAC	: 802.15.4

Jenis antrian : Queue/DropTail/PriQueue

Jenis link layer : LL

Jenis antenna : antena Omni

Maksimal paket pada antrian : 150

Tinggi Antenna : 0,6682 m

Loss : 1 dB (tidak terjadi attenuasi)

Gain Antena : 2,1 dBi

Berdasarkan penelitian [17] proses simulasi perlu diketahui beberapa parameter lain yakni : transmit power output, receive threshold, transmit power, receive power dan initial energy.

3.4.1. Transmit Power Output (Pt)

Transmit power Output (Pt) adalah daya sinyal yang ditransmisikan. Daya sinyal yang dikeluarkan oleh transmitter agar pesan informasi dari transmitter dapat diterima oleh receiver. Disini berdasarkan datasheet Xbee pro transmit power output yang digunakan adalah 60 mW (18 dBm) atau 0,06 W.

3.4.2. Receive Threshold (RxThresh)

Receive Threshold adalah sinyal daya terkecil yang digunakan untuk mendeteksi dan menerima paket yang telah ditransmisikan. Receive threshold menjadikan receiver hanya akan dapat menerima sinyal yang dikirimkan oleh transmitter dengan daya sinyal yang lebih besar dari receive threshold. Disini berdasarkan datasheet Xbee pro memiliki receiver sensitivity sebesar -100 dBm sehingga jika diubah menjadi satuan Watt adalah :

$$\begin{aligned}10 \times \log_{10} (\text{RxThresh}/1 \text{ mW}) &= -100 \\ \log_{10} (\text{RxThresh}/1 \text{ mW}) &= -10 \\ \text{RxThresh}/1\text{mW} &= 10^{-10} \\ \text{RxThresh} &= 10^{-10} \times 1 \text{ mW} \\ &= 10^{-13} \text{ W}\end{aligned}$$

3.4.3. Transmit Power (Tx)

Pada waktu sebuah transmitter akan mengirimkan sinyal informasi menuju receiver, transmitter tersebut akan mengeluarkan daya untuk melakukan transmisi.

Transmit power adalah daya yang dibutuhkan oleh transmitter untuk mentransmisikan sinyal informasi tersebut. Transmit power dapat dituliskan dengan persamaan sebagai berikut.

$$txPower = V_t \times I_t \quad (3.1)$$

Keterangan :

txPower = Transmit Power (dalam Watt)

V_t = Tegangan yang digunakan dalam transmisi (dalam Volt)

I_t = Arus yang digunakan dalam transmisi (dalam Ampere)

Berdasarkan datasheet Xbee pro tegangan yang digunakan bernilai 3,3 V dan transmit current bernilai 250 mA. Maka transmit power yang dihasilkan adalah :

$$\begin{aligned} txPower &= V_t \times I_t \\ &= 3,3 \times 0,25 \\ &= 0,825 \text{ W} \end{aligned}$$

3.4.4. Receive Power (Rx)

Pada waktu sebuah receiver menerima sinyal dari informasi dari transmitter maka terdapat daya yang harus dikeluarkan sebuah receiver tersebut. Maka receive power adalah daya yang dibutuhkan oleh receiver untuk menerima sinyal informasi yang dikirimkan oleh transmitter. Receive power dapat ditulis dengan persamaan berikut.

$$rxPower = V_r \times I_r \quad (3.2)$$

Keterangan :

rxPower = Receive Power (dalam Watt)

V_r = Tegangan yang digunakan dalam penerimaan (dalam Volt)

I_r = Arus yang digunakan dalam penerimaan (dalam Ampere)

Berdasarkan datasheet Xbee pro digunakan tegangan yang bernilai 3,3 V dan receive current bernilai 55 mA. Maka transmit power yang dihasilkan adalah :

$$\begin{aligned}rxPower &= V_r \times I_r \\ &= 3,3 \times 0,055 \\ &= 0,18 \text{ W}\end{aligned}$$

3.4.5. Initial Energy (E)

Initial energy adalah energi yang digunakan dalam perangkat. Energi yang dimaksud adalah besarnya energi yang akan digunakan dalam sebuah perangkat komunikasi. Besarnya energi yang digunakan dalam perangkat adalah bergantung dari tegangan dan arus dari spesifikasi perangkat serta waktu yang dibutuhkan untuk melakukan komunikasi. Intial energy dapat dibuat persamaan yaitu :

$$E = V \times I \times t \quad (3.3)$$

Keterangan :

E = Initial Energy (Joule)

V = Tegangan sumber energi (Volt)

I = Arus dari konstan discharge (Ampere)

t = Waktu (sekon atau detik)

Pada sistem SMKS dalam penelitian ini, baterai yang digunakan adalah baterai *brand* Imedion berkapasitas 9.6 V dan 220 mA untuk 24 jam [26].

Sehingga initial energy-nya adalah :

$$\begin{aligned}E &= V \times I \times t \\ &= 9.6 \times 0,024 \times 24 \times 3600 \\ &= 19906 \text{ J}\end{aligned}$$

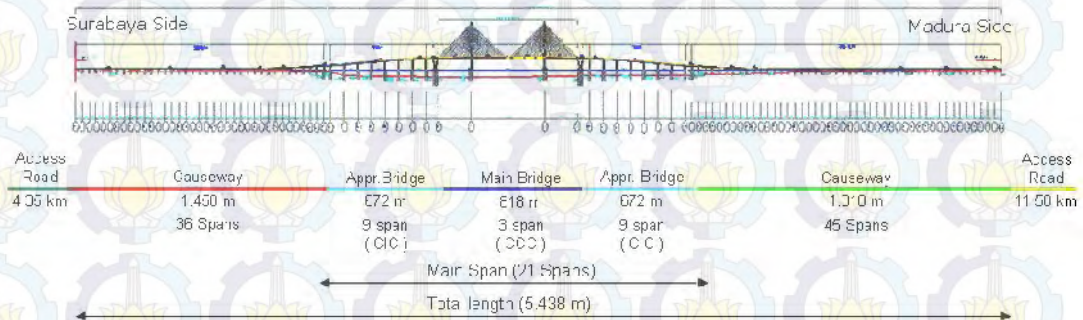
3.5. Desain Sensor pada Jembatan

Dalam peletakan titik-titik sensor pada jembatan harus mengetahui terlebih dahulu bagaimana gambaran jembatan dan kondisi jembatan tersebut. Desain

jembatan dalam simulasi didasarkan pada bentuk asli Jembatan Suramadu dengan memperlihatkan dari sisi samping dan atas.

3.5.1. Jembatan Suramadu

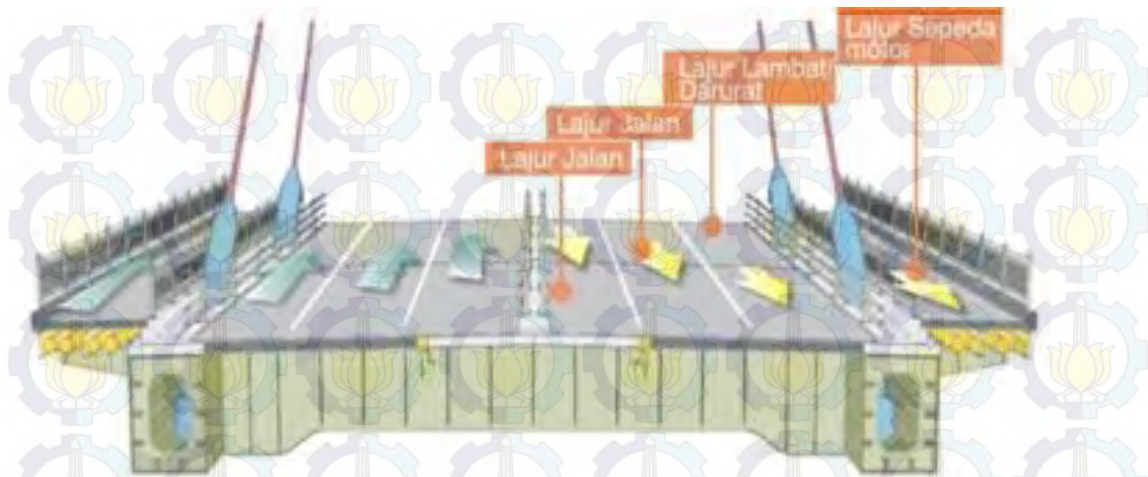
Jembatan Suramadu adalah jembatan yang menghubungkan antara pulau Jawa dan Madura. Pada dasarnya Jembatan Suramadu merupakan gabungan dari tiga jenis jembatan yakni jalan layang atau *causeway*, jembatan penghubung atau *approach bridge* dan jembatan utama atau *main bridge* yang memiliki panjang keseluruhan sepanjang 5.438 meter dengan lebar kurang lebih 30 meter.



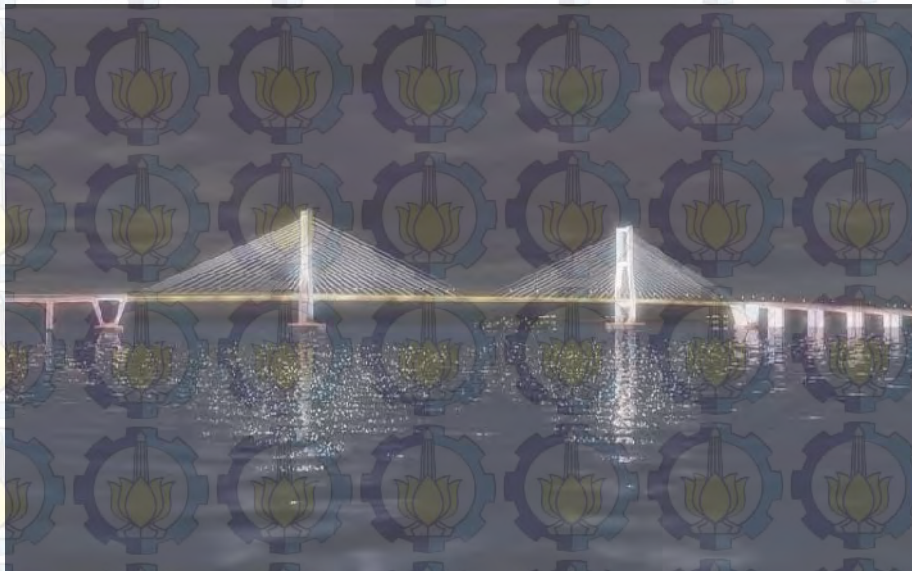
Gambar 3.3 Struktur Jembatan Suramadu

Jembatan ini memiliki spesifikasi pembagian lajur jalan berdasarkan lebar yakni sebagai berikut :

- Lajur jalan utama sebanyak 2 lajur
- Lajur lambat (darurat)
- Lajur sepeda motor



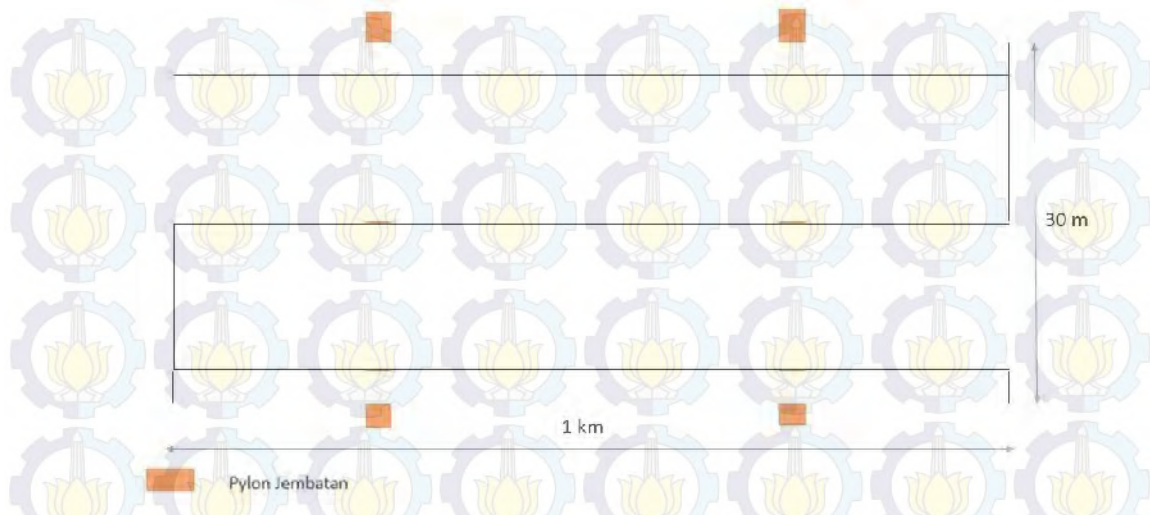
a) SURAMADU tampak atas



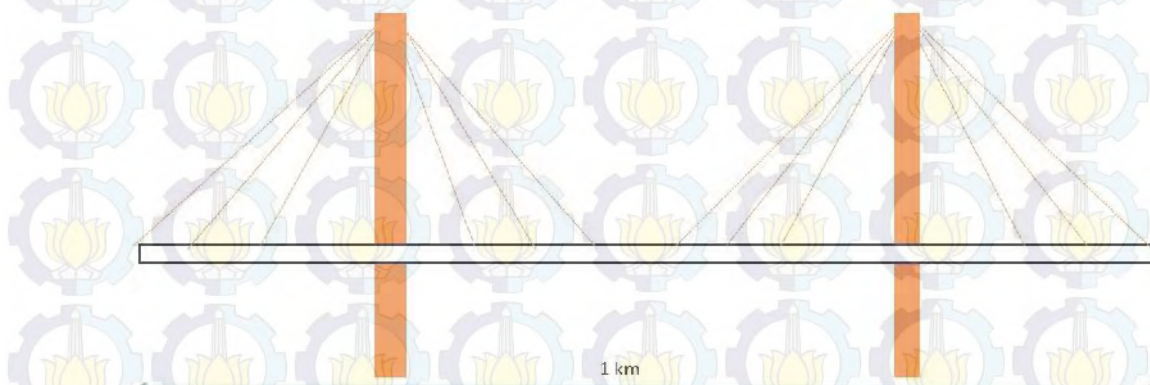
a) SURAMADU tampak samping
Gambar 3.4 Jembatan SURAMADU

3.5.2. Desain Jembatan Dalam Simulasi

Pada simulasi yang akan dilakukan desain jembatan mengacu pada Jembatan Suramadu secara *real*. Jembatan yang akan di simulasikan yakni pada sisi *main bridge*, karena sisi ini merupakan sisi terpenting pada jembatan dan merupakan bagian paling rawan terkena kerusakan. Pada jembatan simulasi diasumsikan panjang jembatan sebesar 1 kilometer dan lebar 30 meter. Berikut pada Gambar 3.5 ditampilkan desain jembatan yang akan digunakan dalam simulasi.



a) Desain jembatan dalam simulasi tampak atas



b) Desain jembatan dalam simulasi tampak samping

Gambar 3.5 Desain jembatan dalam simulasi

3.5.3. Desain peletakan node

Untuk desain peletakan node, node dibagi menjadi dua macam node yakni node koordinator atau *sink* sebagai pengumpul data dari node sensor yang bertransmisi dan node sensor berfungsi sebagai pengolah data pembacaan sensor dan mengirimkan data pada node koordinator. Node koordinator dikondisikan berada di setiap pylon. Selanjutnya untuk asumsi peletakan sensor, terdapat dua macam asumsi. Asumsi pertama jarak antar node berjarak 100 meter dan asumsi kedua jarak antar node 200 meter. Asumsi ini dibuat agar jarak antar node tidak

melebihi jarak maksimum pada NS2 yakni 250 meter. Dan untuk letak pylon 250 dari sisi samping jembatan.

Pada asumsi yang digunakan karakteristiknya didasarkan pada kendaraan yang melewati jembatan. Dimisalkan terdapat mobil yang bergerak dengan kecepatan 80 km/jam maka untuk perhitungan pengiriman paket dirinci sebagai berikut :

- Kecepatan mobil 80 km/jam = 22,22 m/s
- Waktu yang dibutuhkan mobil untuk melewati jembatan = $\frac{1.000}{22,22} = 45$ detik

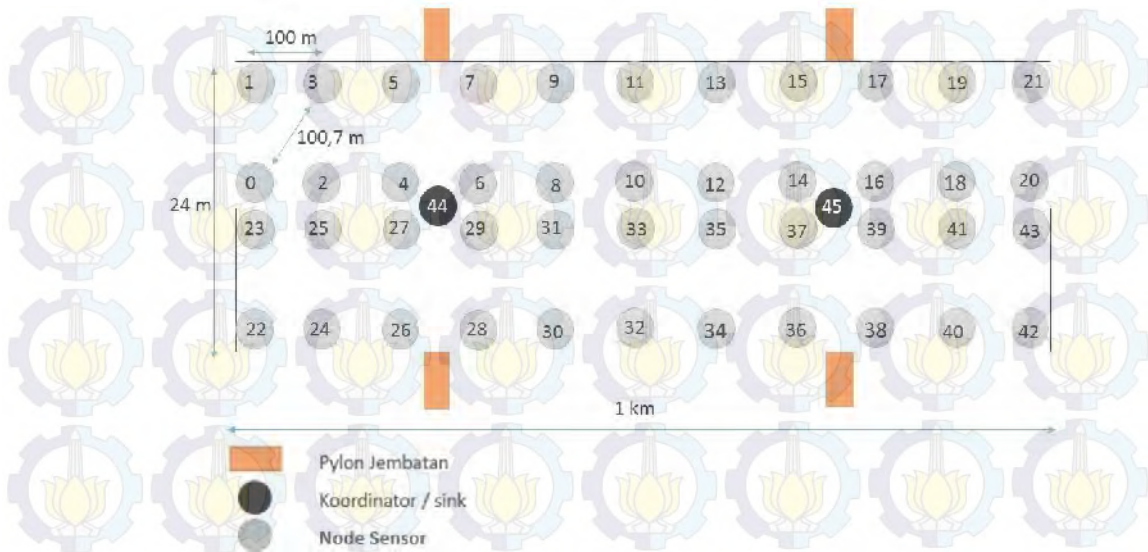
Karena pada desain terdapat dua lajur, yakni lajur dari sisi kiri ke kanan dan lajur dari sisi kanan ke kiri. Pada setiap lajur terdapat 2 node yang diletakkan pada posisi yang bersebrangan, sehingga ketika mobil melaju dan berada pada posisi koordinat yang sama dengan 2 node tersebut maka 2 node tersebut akan mengirimkan data secara bersamaan sehingga dalam 45 detik 2 node akan mengirimkan paket data selama 9 detik. Berikut perhitungannya :

- Waktu pengiriman paket data = $\frac{45}{5} = 9$ detik

Desain asumsi akan diperinci pada penjelasan di bawah ini.

1) Asumsi pertama

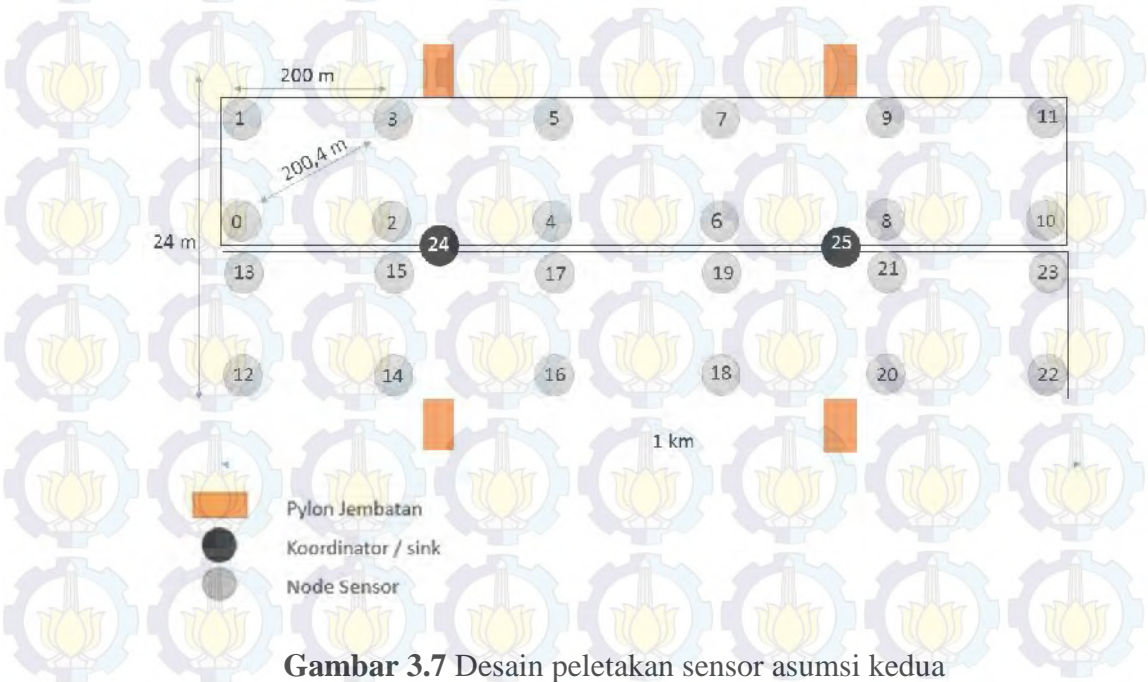
Pada asumsi pertama ini node disebar sebanyak 46 node yang terdiri dari 44 sensor node dan 2 *sink* atau koordinator. Untuk letak node didesain berjarak 100 meter untuk posisi horizontal dan secara vertikal berjarak 12 meter. Sehingga ketika node berkomunikasi secara diagonal jarak antar node sebesar 100,7 meter dan tidak lebih dari jarak maksimumnya.



Gambar 3.6 Desain peletakan sensor asumsi pertama

2) Asumsi kedua

Pada asumsi kedua ini node disebar sebanyak 26 node yang terdiri dari 24 sensor node dan 2 *sink* atau koordinator. Untuk letak node didesain berjarak 200 meter untuk posisi horizontal dan secara vertikal berjarak 12 meter. Sehingga ketika node berkomunikasi secara diagonal jarak antar node sebesar 200,4 meter dan tidak lebih dari jarak maksimumnya. Pada asumsi kedua dapat mengurangi jumlah node yang digunakan, sehingga jumlah node yang digunakan dapat diminimalkan.



Gambar 3.7 Desain peletakan sensor asumsi kedua

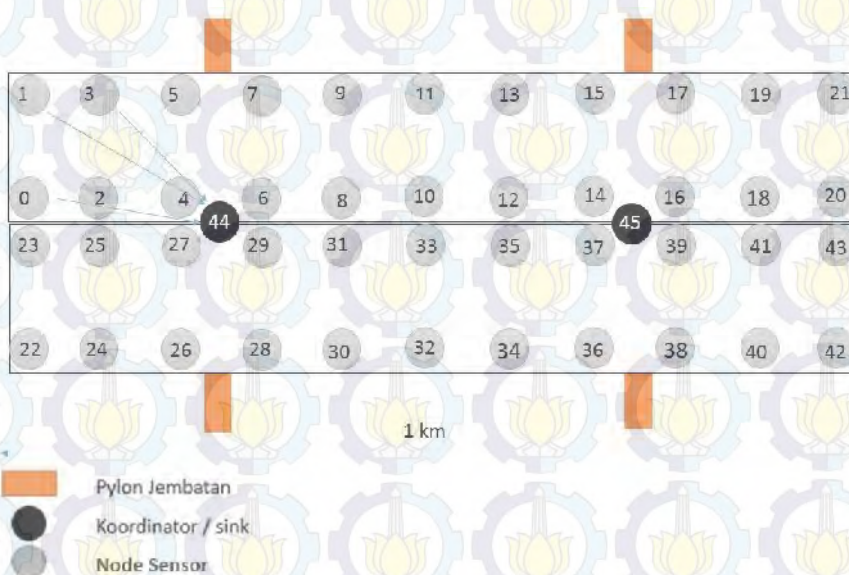
3.6. Desain Topologi

Gambar 3.6 dan Gambar 3.7 menunjukkan desain penyebaran letak sensor pada jembatan dari sudut tampak atas. Dengan mengacu penelitian [17] yakni menggunakan topologi star, mesh dan tree. berdasarkan penelitian [18] [19] sebagaimana telah disebutkan bahwa node di bagi menjadi node sensor dan node koordinator.

Desain topologi yang di tampilkan juga terdapat dua macam berdasarkan desain asumsi yang telah di rancang.

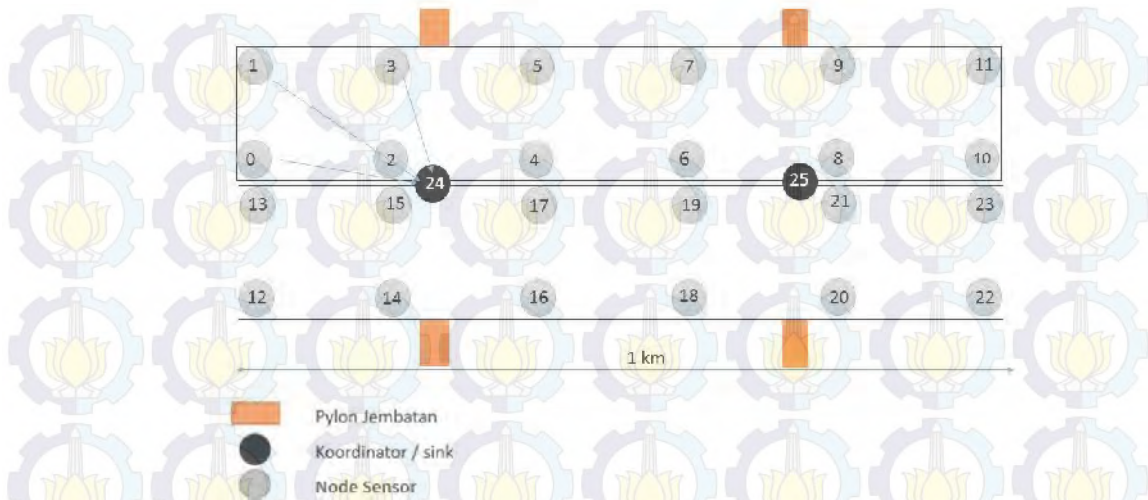
3.6.1. Topologi Star

Topologi Star di desain yakni dengan mengansumsikan node koordinator/*sink* berada di tengah-tengah yakni pada setiap pilon, sedangkan untuk node sensor dapat berkomunikasi secara langsung dengan sink. Desain topologi star dapat dilihat pada Gambar 3.8 dan Gambar 3.9. Berikut pada Gambar 3.8 untuk asumsi pertama dengan jarak antar node sebesar 100 meter.



Gambar 3.8 Desain topologi star asumsi pertama

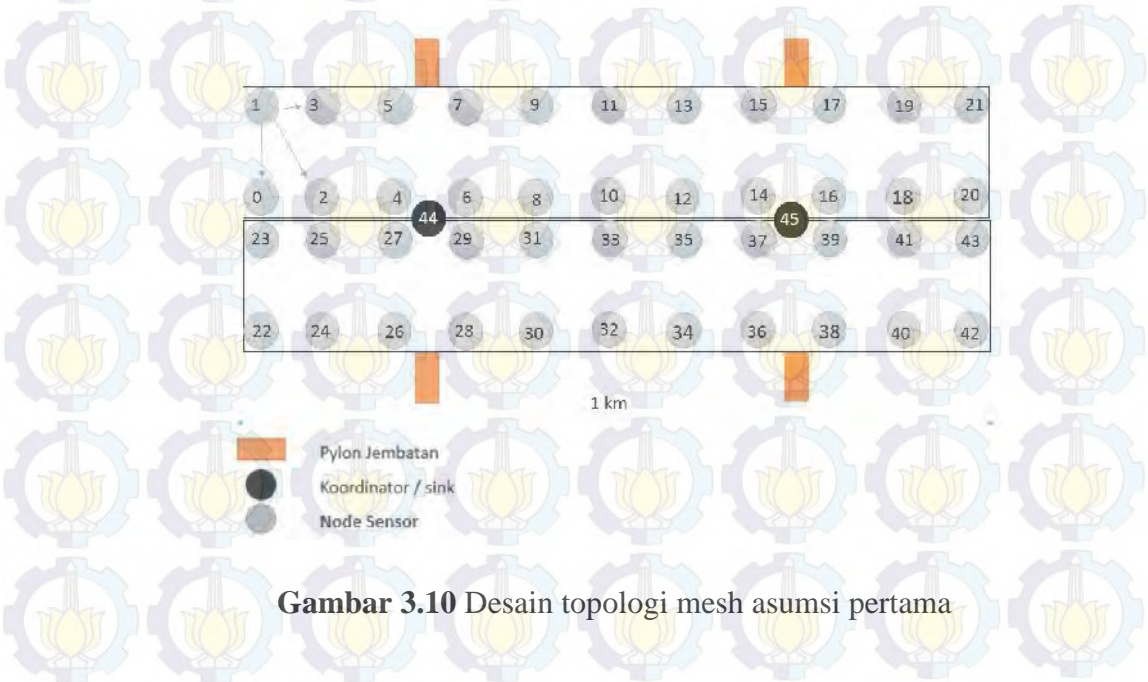
Sedangkan berikut untuk asumsi kedua dengan jarak antar node sebesar 200 meter.



Gambar 3.9 Desain topologi star asumsi kedua

3.6.2. Topologi Mesh

Topologi mesh di desain yakni dengan mengansumsikan node koordinator/*sink* berada di tengah-tengah yakni pada setiap pylon, sedangkan untuk node sensor dapat saling berkomunikasi dimana setiap node sensor memiliki tiga kemungkinan arah komunikasi dengan sensor di sekelilingnya. Desain topologi mesh dapat dilihat pada Gambar 3.10 dan Gambar 3.11. Berikut pada Gambar 3.10 untuk asumsi pertama dengan jarak antar node sebesar 100 meter.



Gambar 3.10 Desain topologi mesh asumsi pertama

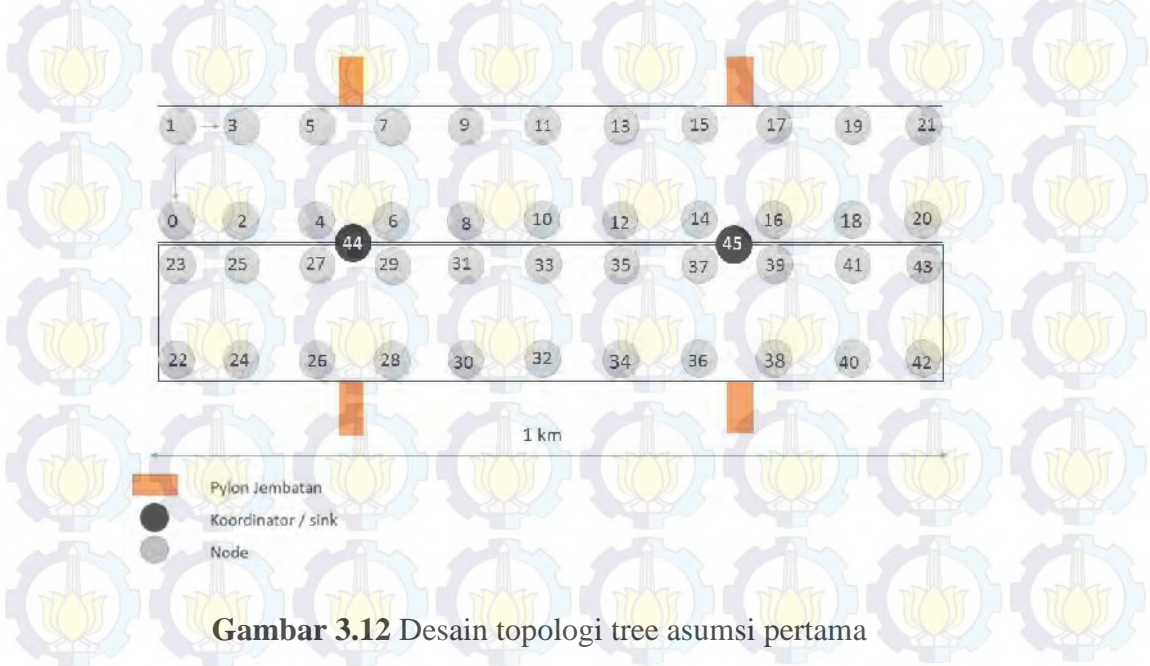
Sedangkan berikut untuk asumsi kedua dengan jarak antar node sebesar 200 meter.



Gambar 3.11 Desain topologi mesh asumsi kedua

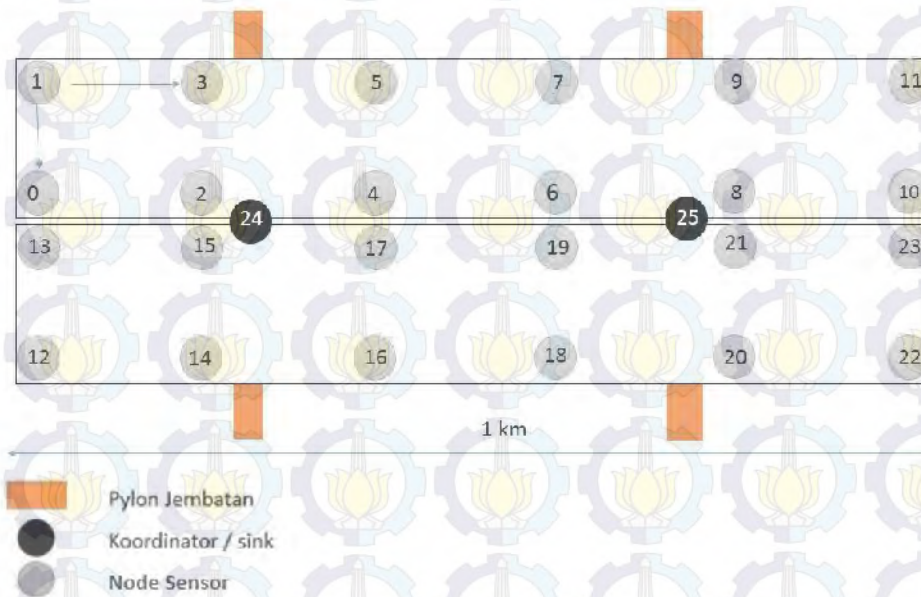
3.6.3. Topologi Tree

Topologi tree di desain yakni dengan mengansumsikan node koordinator/sink berada di tengah-tengah yakni pada setiap pylon, sedangkan untuk node sensor dapat saling berkomunikasi dimana setiap node sensor memiliki dua kemungkinan arah komunikasi dengan sensor di sekelilingnya. Desain topologi mesh dapat dilihat pada pada Gambar 3.12 dan Gambar 3.13. Berikut pada Gambar 3.12 untuk asumsi pertama dengan jarak antar node sebesar 100 meter.



Gambar 3.12 Desain topologi tree asumsi pertama

Sedangkan berikut untuk asumsi kedua dengan jarak antar node sebesar 200 meter.



Gambar 3.13 Desain topologi tree asumsi kedua

3.7. Routing Protokol

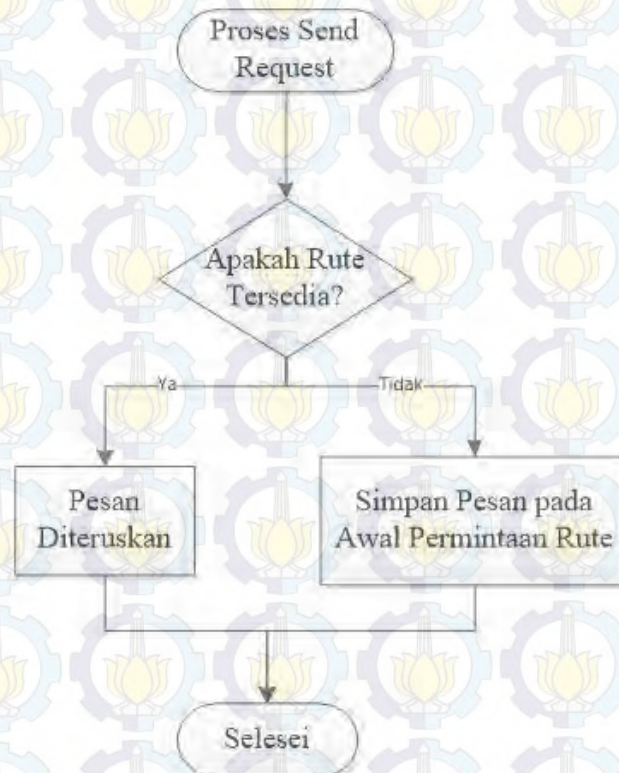
Ada beberapa macam routing protokol yang dipergunakan dalam pengerjaan penelitian ini. Routing protokol tersebut dibagi menjadi dua tipe, yaitu proaktif dan reaktif. Routing protokol bersifat On-demand yang berarti hanya membentuk sebuah rute dari node sumber menuju node tujuan berdasarkan permintaan dari node sumber tersebut. Setiap node menyimpan tabel yang berisi informasi rute ke setiap node yang diketahuinya. Informasi pada setiap node di update jika terjadi perubahan link. Penggunaan protokol routing proaktif secara mendasar memberikan solusi terpendek end-to-end delay, karena informasi routing selalu tersedia dan diperbaharui secara berkala dibandingkan protokol routing reaktif [20].

Routing protokol yang digunakan pada penelitian ini adalah AODV dan DSDV. AODV merupakan salah satu protokol routing reaktif, dimana selama koneksi rute dari pengirim ke penerima telah valid, AODV tidak melakukan pencarian lagi. AODV memelihara rute selama dibutuhkan. Sedangkan DSDV termasuk salah satu protokol proaktif. DSDV juga merupakan salah satu

protokol yang menjaga informasi dalam bentuk tabel (tabel routing) pada setiap node. Tabel tersebut akan sering di update untuk menjaga kekonsistenan dan keakurasian informasi keadaan jaringan. Berikut penjelasan kinerja dari routing protokol AODV dan DSDV.

3.7.1. AODV

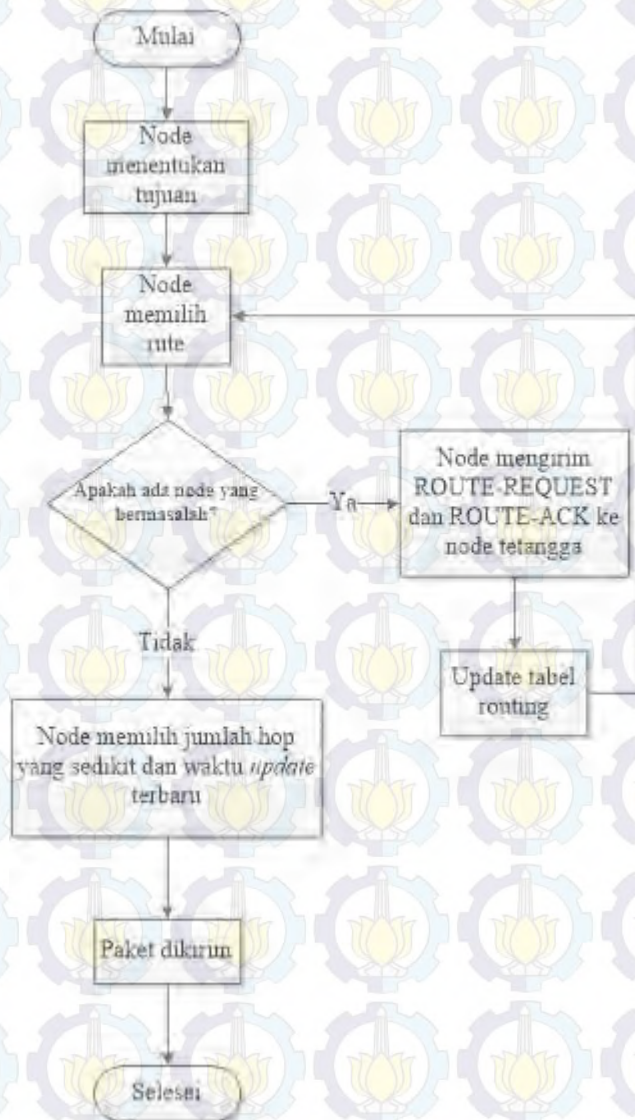
Proses pengiriman paket pada AODV dijelaskan pada Gambar 3.14. Gambar 3.14 menjelaskan proses routing paket pada AODV yang diawali dengan proses permintaan pengiriman update rute. Selanjutnya update rute yang telah dikirim di amati apakah tersedia rute yang bisa dilalui, apabila tidak tersedia maka paket disimpan pada awal permintaan rute namun apabila tersedia rute maka paket diteruskan.



Gambar 3.14 Paket processing AODV [9]

3.7.2. DSDV

Pada DSDV pengiriman paket rendah, faktanya disebabkan karena menggunakan rute link yang rusak [21] [22]. Di DSDV keberadaan stale route atau rute kadaluarsa bukan berarti bahwa tidak ada rute yang valid ke tujuan. Paket dapat diteruskan melalui node tetangga lain yang mungkin memiliki rute ke node tujuan. Pengiriman paket dengan DSDV dapat dilihat pada Gambar 3.15.



Gambar 3.15 Paket *processing* DSDV [9]

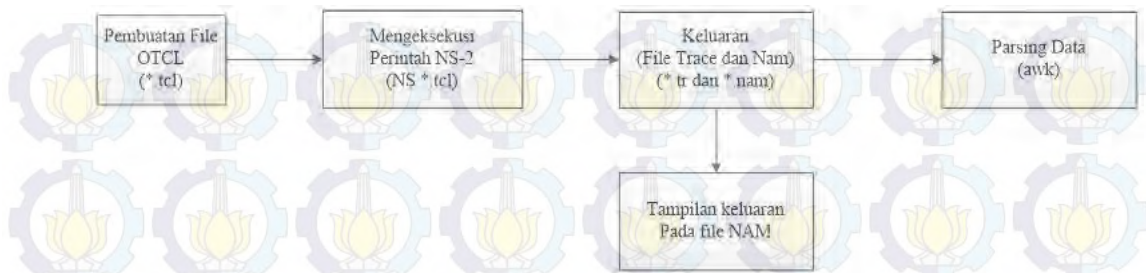
Pada Gambar 3.15 menunjukkan proses kerja dari DSDV. Proses diawali dengan penentuan node sumber dan node tujuan. Selanjutnya node memilih rute yang akan dilewati agar dapat mengirimkan paket sampai ditujuan. Setelah di

dapatkan rute, maka node akan mengecek apakah ada rute yang bermasalah pada suatu node yang akan di lewati. Apabila terdeteksi adanya rute yang bermasalah pada salah satu node yang akan di lewati maka node sumber akan mengirimkan ROUTE-REQUEST dan ROUTE-ACK ke node yang berada disekitar node sumber yang bisasa di sebut node tetangga. Selanjutnya node tetangga akan melakukan update tabel routing yang kemudian dikirimkan kembali ke node sumber. Kemudian node sumber akan melakukan pengecekan kembali terhadap rute yang akan di lewati. Apabila sudah tidak terdeteksi adanya rute yang bermasalah, maka node sumber akan melakukan inisialisasi pada tabel routing yang di kirimkan oleh node tetangga. Node sumber memilih node selanjutnya yang akan di lewati berdasarkan jumlah hop yang sedikit yang waktu update rute terbaru. Setelah di dapatkan node yang pas dengan jumlah hop terkecil, maka paket di kirimkan ke tujuan melalui rute baru yang telah terbentuk

3.8. Model Simulasi

Simulasi dilakukan untuk mengetahui kinerja dari dua routing protokol dengan menggunakan topologi star, mesh dan tree untuk sistem monitoring kesehatan struktur pada jembatan dengan menggunakan NS2. Protokol routing yang digunakan adalah AODV dan DSDV.

Gambar 3.16 menunjukkan keseluruhan proses bagaimana sebuah simulasi jaringan dilakukan dengan NS-2. Keluaran dari NS-2 disebut NAM. NAM menunjukkan pergerakan node dan komunikasi yang terjadi antara node satu dengan node lainnya dalam berbagai kondisi. Dan keluaran lainnya yakni file Trace yang berisi rekaman kejadian yang terjadi yang di simpan pada file *.tr. File keluaran seperti file trace harus diparsing untuk mengekstrak informasi. Parsing dapat dilakukan dengan menggunakan perintah awk. Untuk hasil simulasi di tampilan pada sebuah GUI yang berisi animasi yang di sebut NAM (Network Animator).



Gambar 3.16 Proses simulasi pada NS2 [11]

3.8.1. Format Trace File

Trace file merupakan pencatatan seluruh *event* (kejadian) yang dialami oleh suatu simulasi paket pada simulasi yang dibangun. Pembuatan trace file dilakukan dengan memanggil obyek trace pada library [27]. Gambar 3.17 menunjukkan format isi trace file.

Event	Time	From Node	To Node	Packet Type	Flags	Fid	Source Address	Destination Address	Sequence Number	Packet Id
-------	------	-----------	---------	-------------	-------	-----	----------------	---------------------	-----------------	-----------

Gambar 3.17 Format Trace File

1. Event (kejadian)

Event adalah kejadian yang terjadi dalam simulasi. Kejadian yang dicatat oleh NS yaitu :

r adalah receive yaitu paket yang diterima oleh Node yang dituju

+ adalah enqueue yaitu paket yang masuk dalam antrian atau keluar dari node sumber

- adalah dequeue yaitu paket yang keluar dari antrian

d adalah drop yaitu paket yang di-drop dari antrian

2. Time

Time adalah waktu terjadinya suatu kejadian dalam detik.

3. From Node

From node adalah node sumber.

4. To Node

To node adalah node tujuan.

5. Packet Type

Packet type adalah tipe paket yang dikirim seperti UDP, TCP dan ACK.

6. Packet Size

Packet size adalah ukuran paket dalam byte.

7. Flag

Flag digunakan dalam penanda. Macam-macam flag yang bias digunakan adalah :

E : untuk terjadi kongesti (Congestion Experience/CE)

N : untuk indikasi ECT (ECN-Capable-Transport) pada header IP

C : untuk ECN-Echo

A : untuk pengurangan window kongesti pada header TCP

P : untuk prioritas

F : untuk TCP fast start

8. Fid

Fid adalah penomoran unik dari tiap aliran data

9. Source Address

Source address adalah alamat asal paket.

10. Destination Address

Destination address adalah alamat tujuan paket.

11. Sequence Number

Sequence number adalah nomor urut tiap paket

12. Packet Id

Packet id adalah penomoran unik tiap paket

Berikut contoh hasil trace file dari salah satu asumsi yang dilakukan. Trace file ini nantinya akan di-*parsing* sehingga dapat dianalisa parameter unjuk kinerja dari jaringan yang digunakan.

```
s 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24 0:0 32 0] [0] 0 0  
r 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24 0:0 32 0] [0] 0 0  
s 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24 0:0 32 3] [0] 0 0
```

Gambar 3.18 Contoh trace file pada topologi mesh asumsi pertama

Pada Gambar 3.18, apabila ditulis ulang maka akan nampak seperti berikut

```
s 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy  
19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24  
0:0 32 0] [0] 0 0
```

keterangan :

- s merupakan event / kejadian mengirimkan paket data.
- 0.000000000 merupakan waktu kejadian pengiriman paket data yaitu pada 0.000000000 detik.
- _23_ merupakan node sumber dimana dalam hal ini node 23 merupakan node pengirim paket data.
- AGT merupakan application layer dimana dalam hal ini pada node 23 terjadi pengiriman paket data dari application layer.
- --- menyatakan tidak adanya flag sebagai penanda.
- 0 merupakan penomoran unik dari aliran data.
- CBR merupakan tipe paket yang dikirim oleh application layer node 23.
- 100 merupakan ukuran paket CBR dalam byte yang dikirimkan oleh application layer node 23.
- [0 0 0 0] merupakan penanda bahwa belum terjadi routing.

- [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] menandakan bahwa energi dari node 23 pada saat pengiriman paket data CBR sebesar 100 byte adalah 19906 Joule dengan energi idle yang memiliki nilai 0.000 Joule, energi sleep 0.000 Joule, energi transmit 0.000 Joule, dan energi receive 0.000.
- ----- menandakan tidak terjadi flag.
- [23:24 0:0 32 0] dimana 13:24 menandakan node 23 mengirimkan paket data pada port 0 menuju node 24 dengan port 0, dengan TTL 32 hop, dan 0 menandakan belum ada hop selanjutnya.
- [0] menandakan penomoran unik dari aliran data.
- 0 0 menandakan tahap proses pengiriman node.

3.9. Parameter Kinerja Simulasi

Beberapa parameter yang akan diukur sebagai analisa kinerja sistem yakni :

1. Energi

Energi merupakan kemampuan node saat proses komunikasi data. Energi yang dihasilkan adalah keseluruhan energi yang digunakan baik saat mengirim data ataupun saat membroadcast data dari node tetangganya.

$$\text{Energi} = E_0 - E_t \quad (3.4)$$

Keterangan :

Energi = Energi (Joule)

E_0 = Energi awal sebelum pengiriman paket (Joule)

E_t = Energi akhir setelah pengiriman paket (Joule)

2. Packet Loss

Packet Loss adalah parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang. Kegagalan tersebut dapat disebabkan oleh beberapa kemungkinan, diantaranya yaitu:

- a. Terjadinya *overload* trafik didalam jaringan,
- b. Tabrakan dalam jaringan,
- c. *Error* yang terjadi pada media fisik,
- d. Kegagalan yang terjadi pada sisi penerima antara lain dapat disebabkan karena *overflow* yang terjadi pada *buffer*.

Packet loss dapat dirumuskan seperti pada persamaan :

$$PL = Pl_s - Pl_r \quad (3.5)$$

Keterangan :

Pl = Banyak paket loss (paket)

Pl_s = Banyak paket yang dikirim (paket)

Pl_r = Banyak paket yang diterima (paket)

3. Throughput

Throughput merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut. Throughput juga dapat di definisikan sebagai kecepatan transfer data yang diukur dalam satuan bit per sekon (bps). Throughput dapat dirumuskan seperti pada persamaan:

$$\text{Throughput} = \frac{Pl_r}{t} \quad (3.6)$$

Keterangan :

Throughput = Throughput (bps)

Pl_r = Banyak paket yang diterima (paket)

t = Waktu pengambilan sampel (detik)

4. Delay

Delay atau waktu tunda adalah interval waktu yang dibutuhkan paket data untuk menempuh jarak dari data mulai di kirim sampai dengan data sampai

ditujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, atau juga waktu proses yang lama. *Delay* dinyatakan dalam satuan detik atau *second*. Perhitungan *delay* di dapat dari mengurangkan waktu saat pengiriman paket data dengan waktu saat paket data di terima. *Delay* dapat dirumuskan seperti pada persamaan:

$$\text{Delay} = t_s - t_r \quad (3.7)$$

Keterangan :

Delay = Delay (s)

t_s = Waktu pengiriman paket data (s)

t_r = Waktu penerimaan paketdata (s)

3.10. Parameter Simulasi

Parameter yang digunakan dalam simulasi adalah parameter dengan menggunakan data sheet dari XBeePro series 1 dan disesuaikan dengan software NS-2. Parameter simulasi tersebut dapat dilihat pada Tabel 3.1

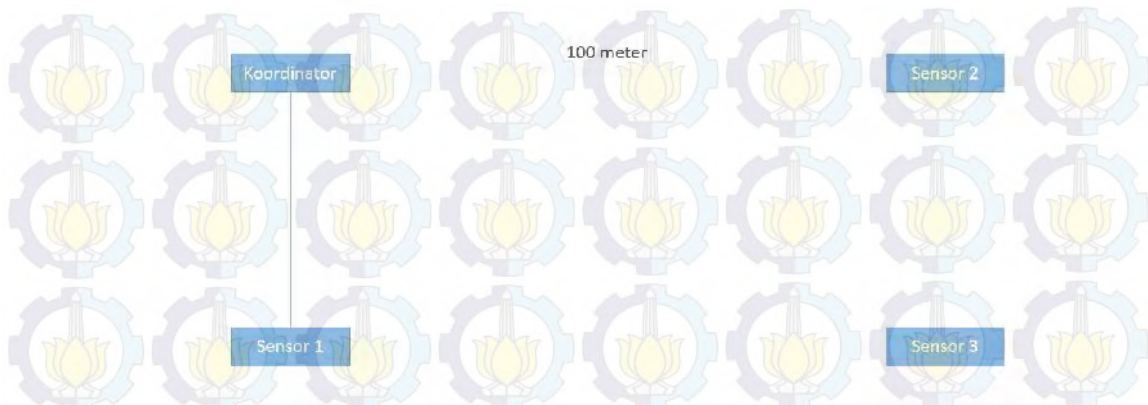
Tabel 3.1 Parameter simulasi

Parameter	Nilai
Jenis Kanal	Wireless Channel
Jenis Layer Fisik dan MAC	802.15.4
Jenis Baterai yang digunakan	Imedion 9.6 V, 220 mA

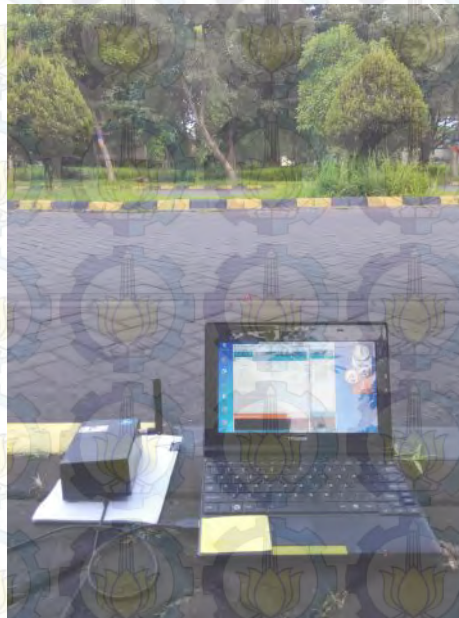
3.11. Validasi Parameter Simulasi

Validasi ini dilakukan untuk memvalidasikan parameter-parameter yang telah diasumsikan dengan mengambil beberapa titik node saja.

Berikut desain untuk validasi parameter simulasi secara *real time*. Pada saat validasi dilakukan ada tiga node sensor dan satu koordinator. Dimana jarak sebesar 12 meter dan 100 meter.



Gambar 3.19 Desain implementasi sistem untuk validasi parameter simulasi dengan jarak 100 meter



Gambar 3.20 Implementasi sistem untuk validasi parameter simulasi

3.11.1. Perangkat Validasi

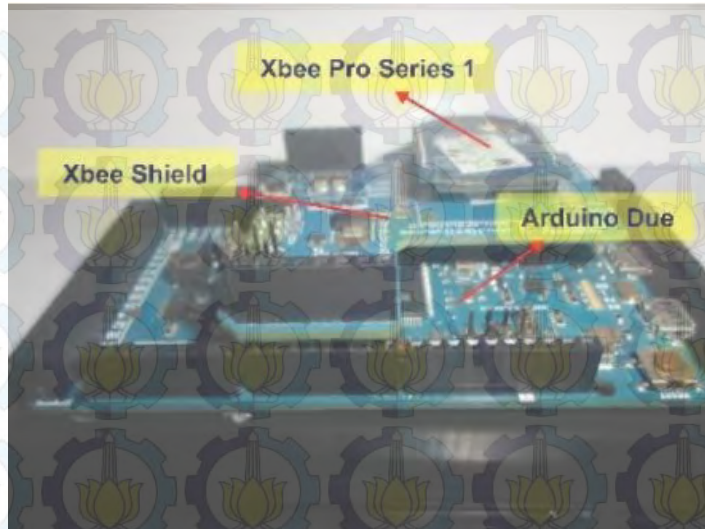
Dalam proses validasi parameter simulasi atau implementasi dari hasil simulasi menggunakan perangkat keras yang terdiri dari dua bagian, yaitu bagian sensing atau node sensor sebagai pengambil data sensor dan bagian pengolahan data atau node koordinator/*sink*. Setiap node menggunakan mikrokontroler arduino duemilanove sebagai pengolah data dan modul komunikasi *Xbee* untuk pentransmisiannya. Sedangkan untuk proses sensing menggunakan sensor accelerometer MMA 7361.



Gambar 3.21 Bentuk fisik node

➤ **Node Koordinator**

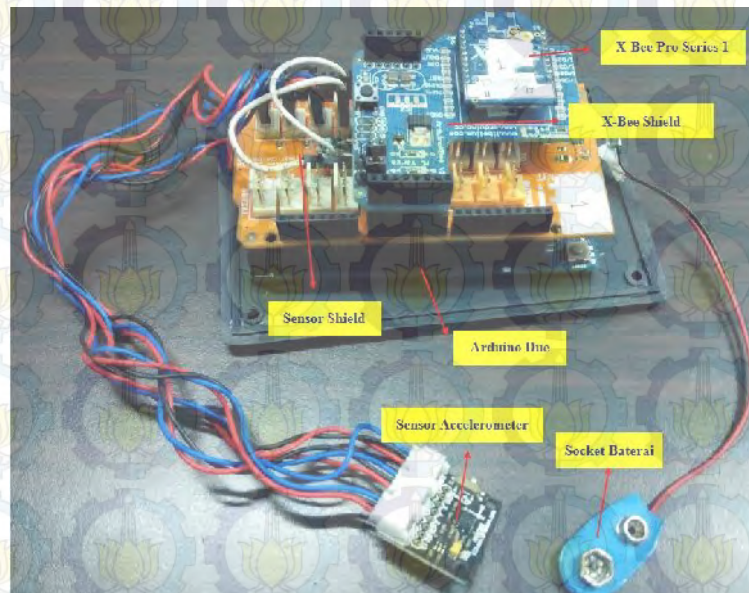
Node koordinator sebagai penerima data dan pengolah data dari node sensor serta yang selanjutnya akan dikirimkan ke *base station*. Node koordinator terdiri dari mikrokontroler arduino due sebagai pengolah data serta *Xbee* sebagai modul komunikasi untuk dapat berkomunikasi dengan perangkat pada sisi node sensor. Pada sisi koordinator tidak menggunakan ataupun terhubung dengan sensor karena pada sisi koordinator ini hanya berfungsi sebagai penerima data dari node sensor. Pada node koordinator menggunakan mikrokontroler arduino duemilanove sebagai pengolah data dan modul komunikasi xbee untuk dapat terhubung secara nirkabel dengan node sensor. Selanjutnya, modul *Xbee* sebagai modul komunikasi data. *Xbee* yang digunakan adalah *Xbee Pro Series 1*. Modul ini adalah perangkat dengan protokol standart IEEE 802.15.4. *Xbee Pro Series 1* memiliki kemampuan berkomunikasi secara *point-to-point*, dan *point-to-multipoint*. Pada bagian node sensor modul *Xbee* diatur agar dapat melakukan komunikasi data secara *multipoint-to-point*.



Gambar 3.22 Susunan node koordinator.

➤ **Node Sensor**

Node sensor terdiri dari sensor accelerometer sebagai alat sensing, mikrokontroler arduino due sebagai pengolah data serta Xbee sebagai modul komunikasi untuk dapat terhubung dengan perangkat pada sisi node koordinator.



Gambar 3.23 Susunan node sensor.



(HALAMAN INI SENGAJA DIKOSONGKAN)

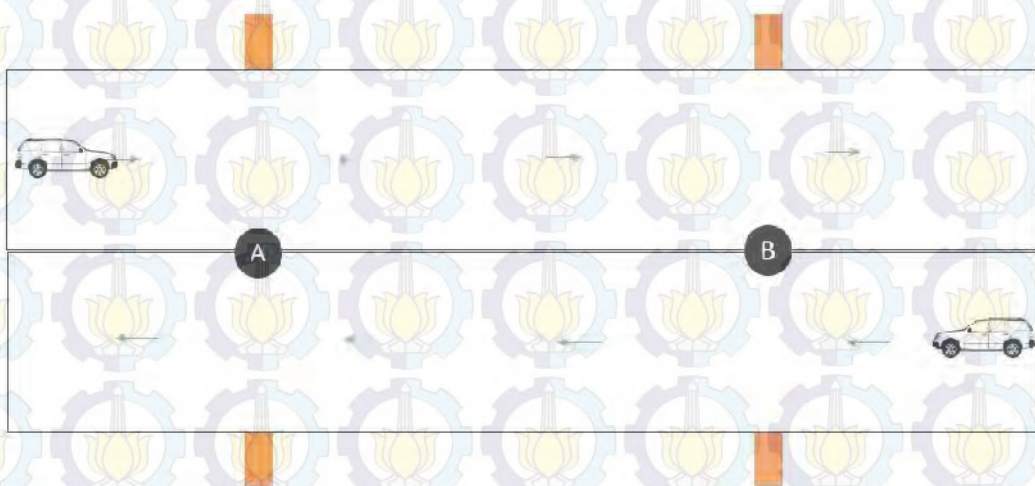
BAB 4 HASIL DAN ANALISA

Pada bab ini membahas mengenai pelaksanaan pengujian yang dilakukan dengan menggunakan skenario yang telah dijelaskan pada bab sebelumnya. Pengujian diawali dengan skenario pengujian terhadap asumsi yang ditetapkan.

4.1. Skenario Pengujian Sistem

Dalam pengujian sistem disini ditekankan pada dua jalur, yakni pada jalur kendaraan roda empat atau lebih baik sisi arah ke madura maupun sisi arah ke surabaya. Skenario yang dirancang menggunakan dua macam skenario yakni skenario dengan asumsi pertama menggunakan jarak 100 meter x 12 meter untuk setiap jalur dan skenario dengan asumsi kedua menggunakan jarak 200 meter x 12 meter untuk setiap jalur. Setiap asumsi juga membandingkan penggunaan routing dan topologi. Untuk routing menggunakan dua macam routing, yakni routing AODV dan DSDV. Sedangkan untuk topologi menggunakan topologi star, tree dan mesh.

Pada skenario pengujian sistem, jembatan dibagi menjadi dua jalur, dimana satu sisi untuk jalur perjalanan kendaraan dari kiri ke kanan dan sisi lainnya untuk jalur perjalanan kendaraan kanan ke kiri.



Gambar 4.1. Pembagian arah jalur pada jembatan

Pembagian rute dan penentuan node koordinator yakni untuk lajur kiri ke kanan menggunakan node koordinator B, sedangkan untuk lajur kanan ke kiri menggunakan node koordinator A.

Pada setiap asumsi yang akan digunakan karakteristiknya didasarkan pada kendaraan yang melewati jembatan. Dimisalkan terdapat mobil yang bergerak dengan kecepatan 80 km/jam maka untuk perhitungan pengiriman paket dirinci sebagai berikut :

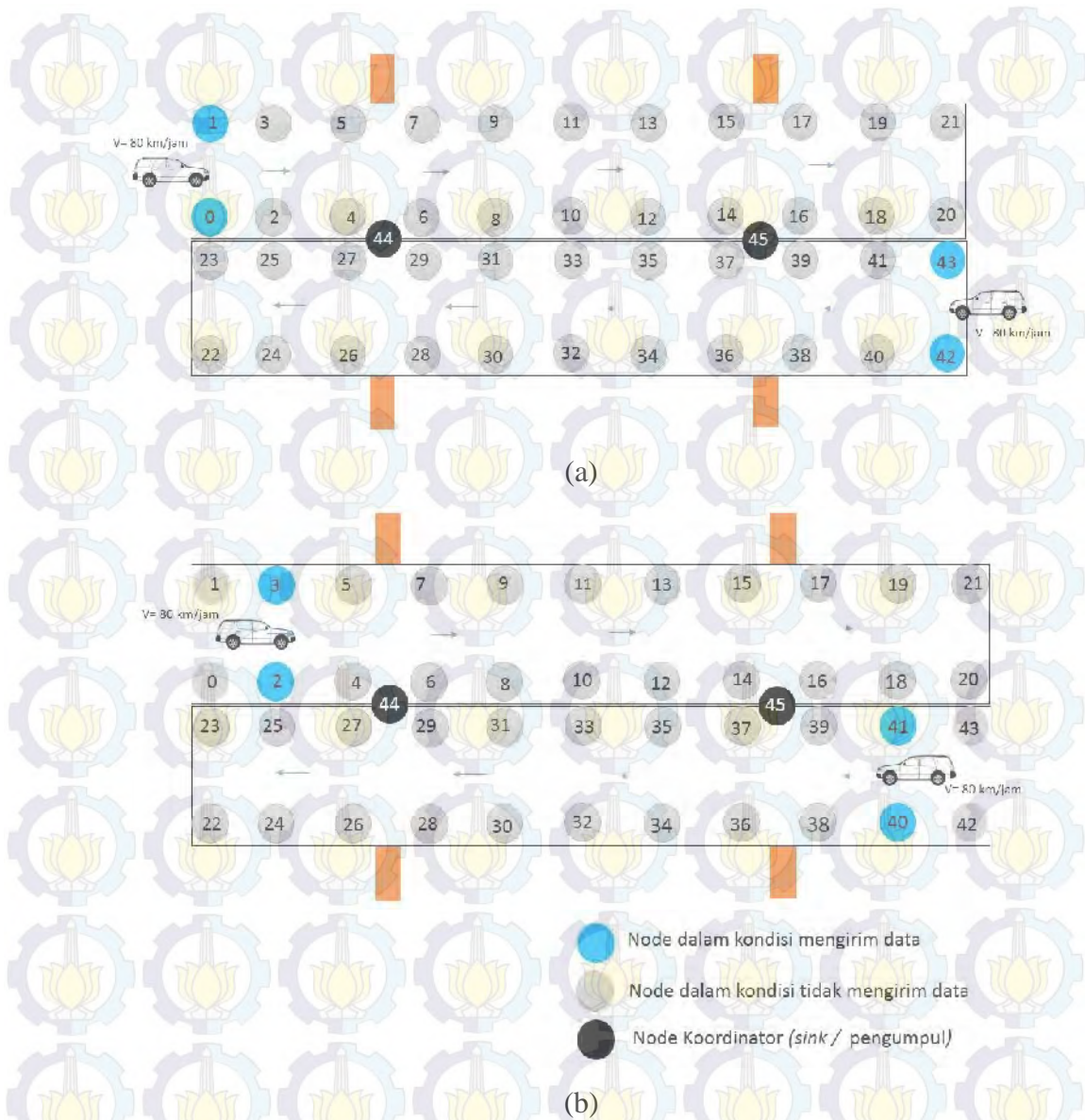
- Kecepatan mobil 80 km/jam = 22,22 m/s
- Waktu yang dibutuhkan mobil untuk melewati jembatan = $\frac{1.000}{22,22} = 45$ detik

Karena pada desain terdapat dua lajur, yakni lajur dari sisi kiri ke kanan dan lajur dari sisi kanan ke kiri. Pada setiap lajur terdapat 2 node yang diletakkan pada posisi yang bersebrangan, sehingga ketika mobil melaju dan berada pada posisi koordinat yang sama dengan 2 node tersebut maka 2 node tersebut akan mengirimkan data secara bersamaan sehingga dalam 45 detik 2 node akan mengirimkan paket data selama 9 detik. Berikut perhitungannya :

- Waktu pengiriman paket data = $\frac{45}{5} = 9$ detik

4.2. Asumsi Pertama

Pada asumsi pertama ini, posisi node di kondisikan jarak antar node terletak dengan jarak 100 meter dan 12 meter. Pada asumsi ini di skenarioikan terdapat sebuah kendaraan yang melintas dengan kecepatan 80 km/jam, sehingga node-node sensor akan mengikuti pergerakan kendaraan dan node yang akan dilewati akan bersiap-siap untuk mengirimkan paket data menuju node koordinator.



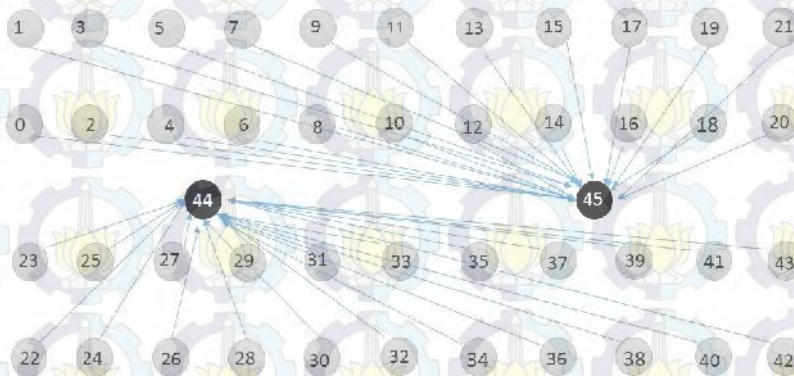
Gambar 4.2. Skenario pada asumsi pertama

Terlihat dari Gambar 4.2 dimana untuk skenario pada asumsi pertama. Pada sisi lajur kiri ke kanan ketika ada kenjaraan melaju memasuki jembatan sehingga posisi koordinatnya kendaraan sama dengan node 0 dan node 1 maka node 0 dan node 1 akan mengirim paket data menuju node 45 (node koordinator). Sedangkan Pada sisi lajur kanan ke kiri ketika ada kenjaraan melaju memasuki jembatan maka node 42 dan node 43 akan mengirim paket data menuju node 44 (node koordinator) seperti pada Gambar 4.2 (a). Node aktif mengirimkan data ditandai dengan node berwarna biru. Pada sisi lajur kiri ke kanan ketika posisi kendaraan

sama dengan node 2 dan node 3 maka node 2 dan node 2 akan mengirim paket data menuju node 45, untuk node 0 dan node 1 sudah tidak lagi mengirimkan paket data. Sedangkan Pada sisi lajur kanan ke kiri ketika posisi kendaraan sama dengan node 40 dan node 41 maka node 40 dan node 41 akan mengirim paket data menuju node 44, untuk node 42 dan node 43 sudah tidak lagi mengirimkan paket data seperti pada Gambar 4.2 (b).

4.2.1. Topologi Star

Topologi star merupakan topologi paling sederhana dimana komunikasi terjadi secara *point-to-point* node dengan sink. Desain topologi seperti Gambar 4.3.



Gambar 4.3. Topologi star asumsi pertama

Terlihat pada Gambar 4.3, bahwa pada topologi star setiap node berkomunikasi secara langsung dengan node koordinator (node 44 dan node 45).

Topologi pada asumsi yang ini mirip dengan Gambar 4.2 dimana setiap node mengirim paket data langsung menuju koordinator. Pada sisi kiri ke kanan awalnya, node 0 dan node 1 akan mengirimkan paket data secara bersamaan dan langsung menuju node 45 selama 9 detik, kemudian node 2 dan node 3 akan mengirimkan paket data secara bersamaan dan langsung menuju node 45 selama 9 detik. Dilanjutkan node 4 dan node 5 mengirimkan paket data, lalu node 6 dan node 7, dan yang terakhir adalah node 20 dan node 21 mengirimkan paket data secara bersamaan langsung menuju node 45. Sedangkan Pada sisi kanan ke kiri awalnya, node 42 dan node 43 akan mengirimkan paket data secara bersamaan

dan langsung menuju node 44 selama 9 detik, kemudian node 40 dan node 41 akan mengirimkan paket data secara bersamaan dan langsung menuju node 45 selama 9 detik. Dilanjutkan node 38 dan node 39 mengirimkan paket data, lalu node 36 dan node 37, dan yang terakhir adalah node 22 dan node 23 mengirimkan paket data secara bersamaan langsung menuju node 44. Hasil dari simulasi tampak pada Tabel 4.1. dan Tabel 4.2. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

Tabel 4.1. Hasil simulasi topologi star dengan routing DSDV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1.	<i>Throughput</i>	0.65 kbps
2.	<i>Delay end-to-end</i> rata-rata	0.840 ms
3.	<i>Packet Loss</i>	139 paket
4.	Energi <i>end-to-end</i> rata-rata	0.106 Joule

Tabel 4.2. Hasil simulasi topologi *star* dengan routing AODV dengan asumsi pertama

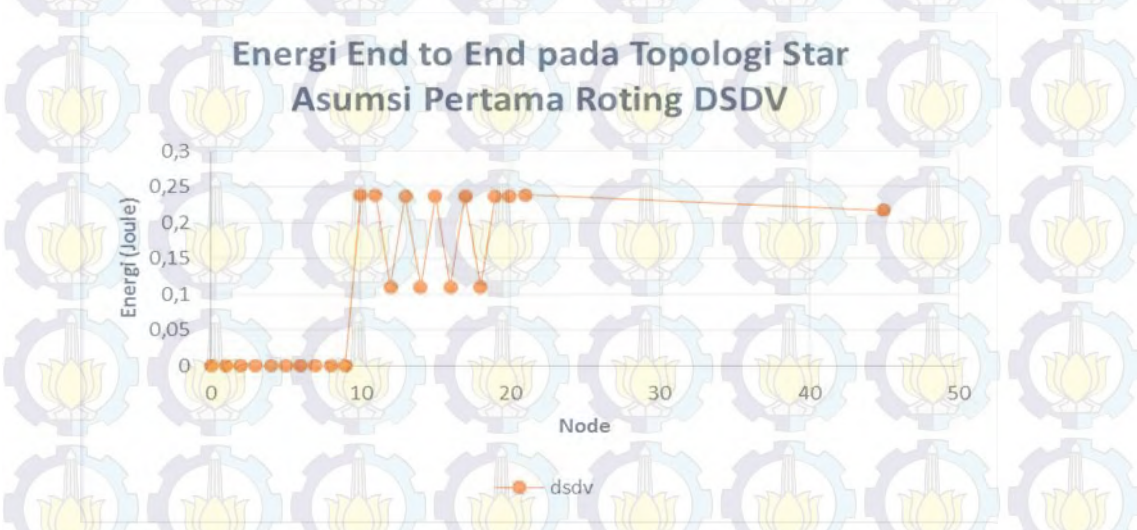
No.	Parameter Kinerja Jaringan	Nilai
1)	<i>Throughput</i>	0.69 kbps
2)	<i>Delay end-to-end</i> rata-rata	0.532 ms
3)	<i>Packet Loss</i>	142 paket
4)	Energi <i>end-to-end</i> rata-rata	0.154 Joule

karena terdapat node yang melebihi jarak jangkauan maka terdapat beberapa paket yang tidak dikirimkan.

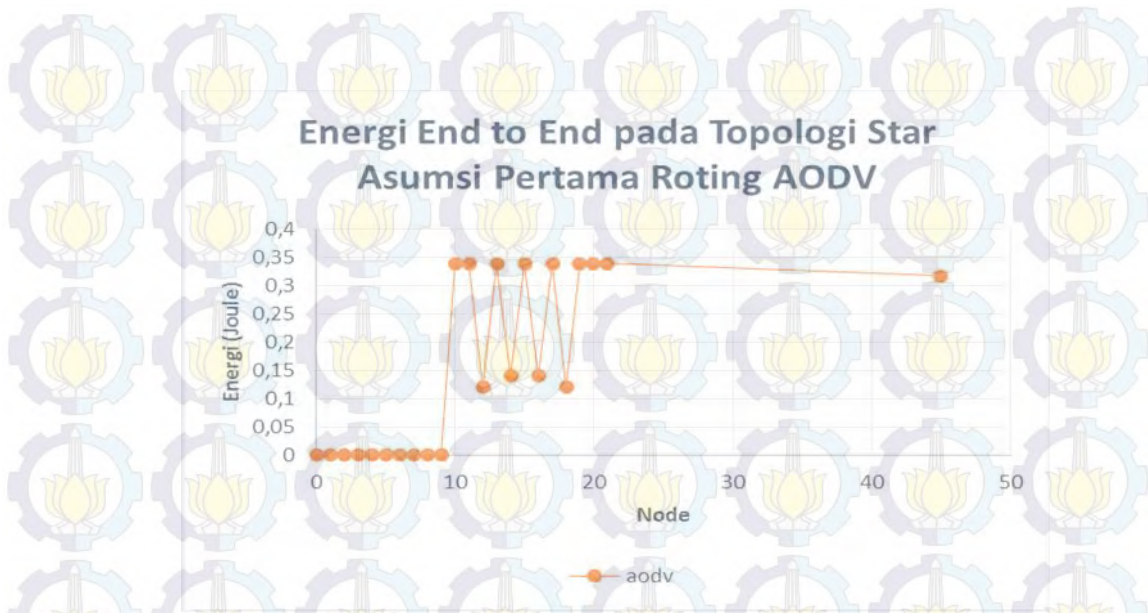
Parameter energi *end-to-end* pada topologi star asumsi pertama untuk routing AODV dan DSDV pada masing-masing node ditunjukkan pada Gambar 4.4 dan Gambar 4.5. Pada lajur kanan ke kiri node 0, node 1, node 2,

node 3, node 4, node 5, node 6, node 7, dan node 8, energi *end-to-end* masing-masing node tersebut adalah nol. Hal ini disebabkan ketika node-node tersebut mengirimkan paket data secara langsung menuju node 0, karena jarak yang melebihi jangkauan XBee maka tidak paket tersebut tidak dikirimkan. Sedangkan pada node 10, node 11, node 12, node 13, node 15, node 17, node 19, node 21, dan node 20, masing-masing memiliki nilai energi *end-to-end* untuk routing AODV berturut-turut 0,338 Joule, 0,339 Joule, 0,338 Joule, 0,338 Joule, 0,338 Joule, 0,338 Joule, 0,339 Joule dan 0,338 Joule, Node – node ini untuk aktifitas “mendengar”. Node 12 dan Node 18 memiliki nilai energi *end-to-end* 0,12. Node 14 dan Node 16 memiliki nilai energi *end-to-end* 0,14 Joule untuk aktifitas pengiriman paket data menuju node 45 (Koordinator). Sedangkan node 45 memiliki nilai energi *end-to-end* 0,317 Joule untuk aktifitas menerima paket data dan “mendengar. Sedangkan untuk routing DSDV berturut-turut 0,238 Joule, 0,239 Joule, 0,238 Joule, 0,238 Joule, 0,238 Joule, 0,238 Joule, 0,239 Joule dan 0,238 Joule. Node 12, Node 14, Node 16 dan Node 18 memiliki nilai energi *end-to-end* 0,11 untuk aktifitas pengiriman paket data menuju node 45 (Koordinator). Sedangkan node 45 memiliki nilai energi *end-to-end* 0,217 Joule untuk aktifitas menerima paket data dan “mendengar.

Berdasarkan Gambar 4.4 dan Gambar 4.5 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 0,154 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 0,106 Joule.



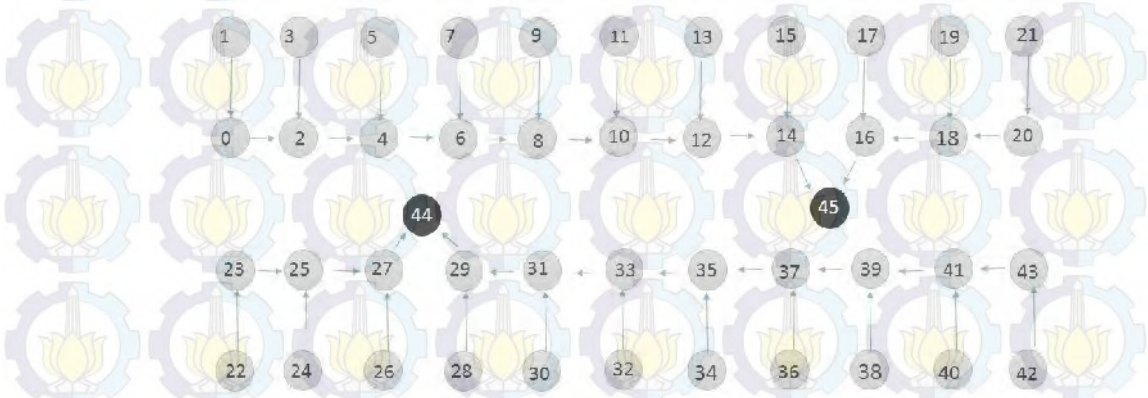
Gambar 4.4. Energi End-to-end topologi star asumsi pertama routing DSDV



Gambar 4.5. Energi End-to-end topologi star asumsi pertama routing AODV

4.2.2. Topologi Mesh

Topologi Mesh atau biasa disebut komunikasi *peer-to-peer* adalah komunikasi yang terjadi antar node Desain topologi seperti Gambar 4.6.



Gambar 4.6. Topologi mesh asumsi pertama

Terlihat pada Gambar 4.6 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya.

Pada skenario dengan asumsi pertama ini, pada sisi kiri ke kanan, ketika sebuah mobil memiliki koordinat yang sama dengan node 1 dan node 0, maka node 1 mengirimkan data melalui node 0 yang kemudian di teruskan dari node 0

menuju node 2. Node 2 juga menerima data dari node 3 dan seterusnya sehingga data sampai di node 45. Sedangkan pada sisi kanan ke kiri, ketika sebuah mobil memiliki koordinat yang sama dengan node 42 dan node 43, maka node 42 mengirimkan data melalui node 43 yang kemudian diteruskan dari node 43 menuju node 41. Node 41 juga menerima data dari node 40 dan seterusnya sehingga data sampai di node 44.

Hasil dari simulasi tampak pada Tabel 4.3. dan Tabel 4.4. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

Tabel 4.3. Hasil simulasi topologi mesh dengan routing DSDV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	7.11 kbps
2	<i>Delay end-to-end</i> rata-rata	27,25 ms
3	<i>Packet Loss</i>	2 paket
4	Energi <i>end-to-end</i> rata-rata	5,09 Joule

Tabel 4.4. Hasil simulasi topologi mesh dengan routing AODV dengan asumsi pertama

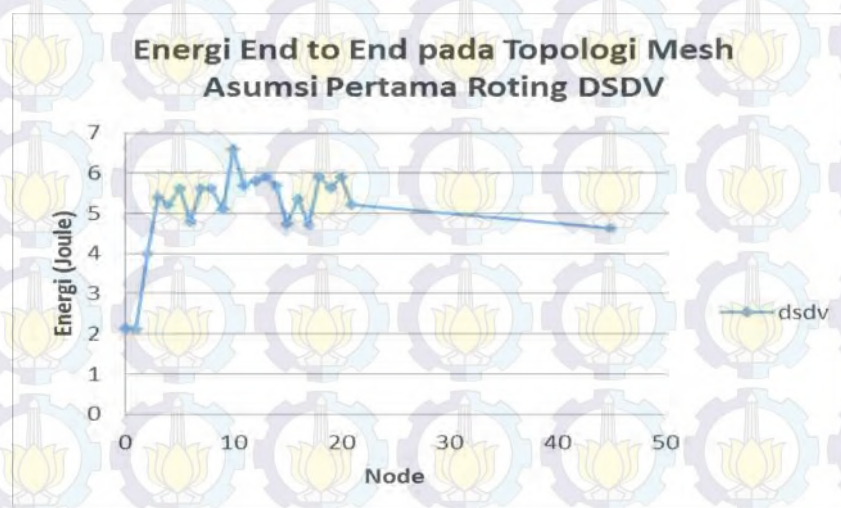
No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	21.28 kbps
2	<i>Delay end-to-end</i> rata-rata	10.68 ms
3	<i>Packet Loss</i>	1 paket
4	Energi <i>end-to-end</i> rata-rata	5.66 Joule

Terlihat pada Gambar 4.7, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,14 Joule, node 1 bernilai 2.11 Joule, node 2 bernilai 4 Joule, node 3 bernilai 5.4 Joule, node 4 bernilai 5.2 Joule, node 5 bernilai 5.6 Joule, node 6 bernilai 4.8 Joule, node 7

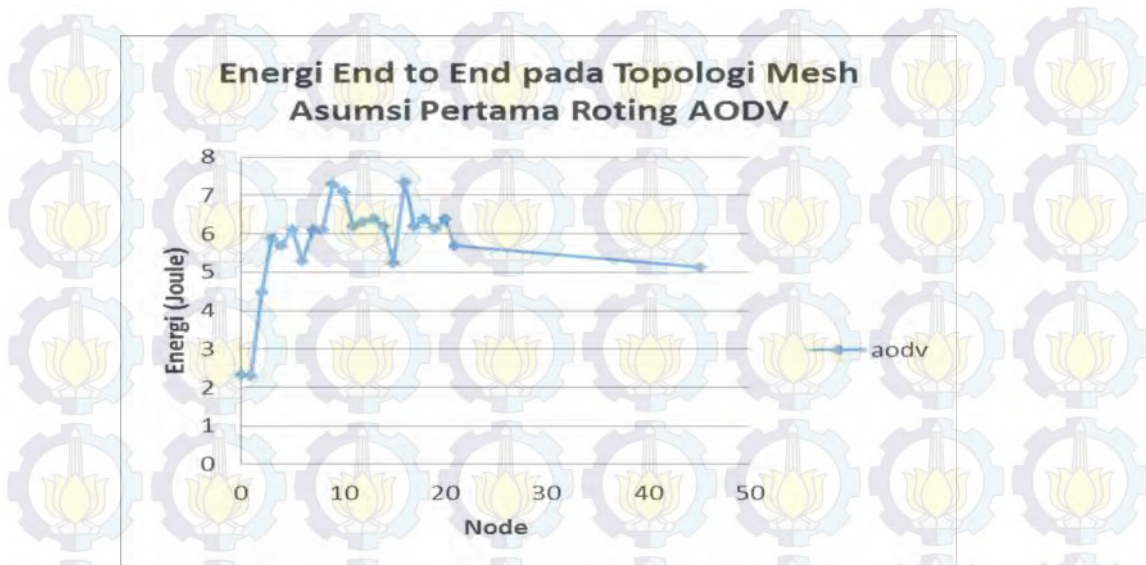
bernilai 5.6 Joule, node 8 bernilai 5.6 Joule, node 9 bernilai 5.1 Joule, node 10 bernilai 6.6 Joule., node 11 bernilai 5.7 Joule, node 12 bernilai 5.8 Joule, node 13 bernilai 5.9 Joule, node 14 bernilai 4.74 Joule, node 15 bernilai 5,34 Joule dan node 16 bernilai 4.7 Joule, node 17 bernilai 5.9 Joule, node 18 bernilai 5.63 Joule, node 19 bernilai 5.9 Joule, node 20 bernilai 5.4 Joule, node 21 bernilai 5.2 Joule dan node 45 bernilai 4.63 Joule.

Terlihat pada Gambar 4.8, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2.34 Joule, node 1 bernilai 2.31 Joule, node 2 bernilai 4.5 Joule, node 3 bernilai 5.9 Joule, node 4 bernilai 5.7 Joule, node 5 bernilai 6.1 Joule, node 6 bernilai 5.3 Joule, node 7 bernilai 6.1 Joule, node 8 bernilai 6.1 Joule, node 9 bernilai 7.3 Joule, node 10 bernilai 7.1 Joule., node 11 bernilai 6.2 Joule, node 12 bernilai 6.3 Joule, node 13 bernilai 6.4 Joule, node 14 bernilai 6.2 Joule, node 15 bernilai 5,24 Joule dan node 16 bernilai 5,34 Joule, node 17 bernilai 6.2 Joule, node 18 bernilai 6.4 Joule, node 19 bernilai 6.13 Joule, node 20 bernilai 6.4 Joule, node 21 bernilai 65.7 Joule dan node 45 bernilai 5.13 Joule.

Berdasarkan Gambar 4.7 dan Gambar 4.8 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 5.67 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 5.09 Joule.



Gambar 4.7. Energi End-to-end topologi mesh asumsi pertama routing DSDV



Gambar 4.8. Energi End-to-end topologi mesh asumsi pertama routing AODV

4.2.3. Topologi Tree

Desain topologi tree dapat dilihat pada Gambar 4.9 berikut ini.



Gambar 4.9. Topologi tree asumsi pertama

Terlihat pada Gambar 4.9 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya. Pada topologi ini hampir sama dengan topologi mesh, namun pada topologi ini komunikasi node terbagi atas beberapa jalur untuk sampai ke koordinator

Untuk sisi kiri ke kanan, jalur pertama yakni dari node 1 menuju node 3, node 5, node 7, node 9, node 11, node 13, node 15, node 14 baru ke node 45. Jalur kedua yakni node 0 menuju node 2, node 4, node 6, node 8, node 10, node 12, node 14, baru ke node 45. Jalur ke tiga dari node 17 ke node 19, node 21, node 20, node 18, node 16 baru ke node 45. Sedangkan sisi kanan ke kiri, jalur pertama yakni dari node 43 menuju node 41, node 39, node 37, node 35, node 33, node 13, node 31, node 29 baru ke node 44. Jalur kedua yakni node 42 menuju node 40, node 38, node 34, node 32, node 30, node 28, node 29, baru ke node 44. Jalur ke tiga dari node 26 ke node 24, node 22, node 23, node 25, node 27 baru ke node 44. Hasil dari simulasi tampak pada Tabel 4.5. dan Tabel 4.6. Hasil simulasi di ambil dari nilai rata-rata kedua sisi.

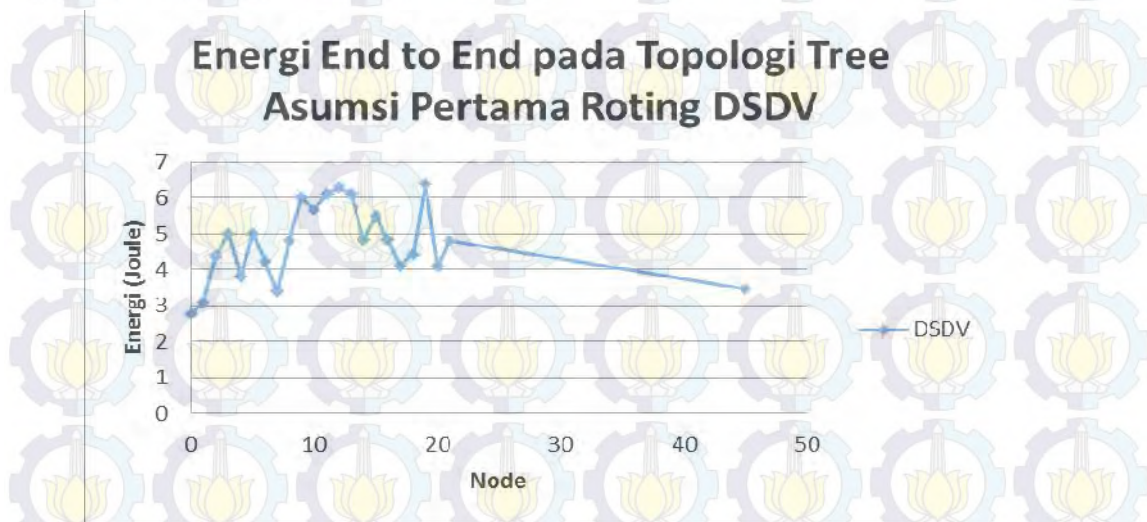
Tabel 4.5. Hasil simulasi topologi tree dengan routing DSDV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	6.35 kbps
2	<i>Delay end-to-end</i> rata-rata	26,34 ms
3	<i>Packet Loss</i>	2 paket
4	Energi <i>end-to-end</i> rata-rata	4.74 Joule

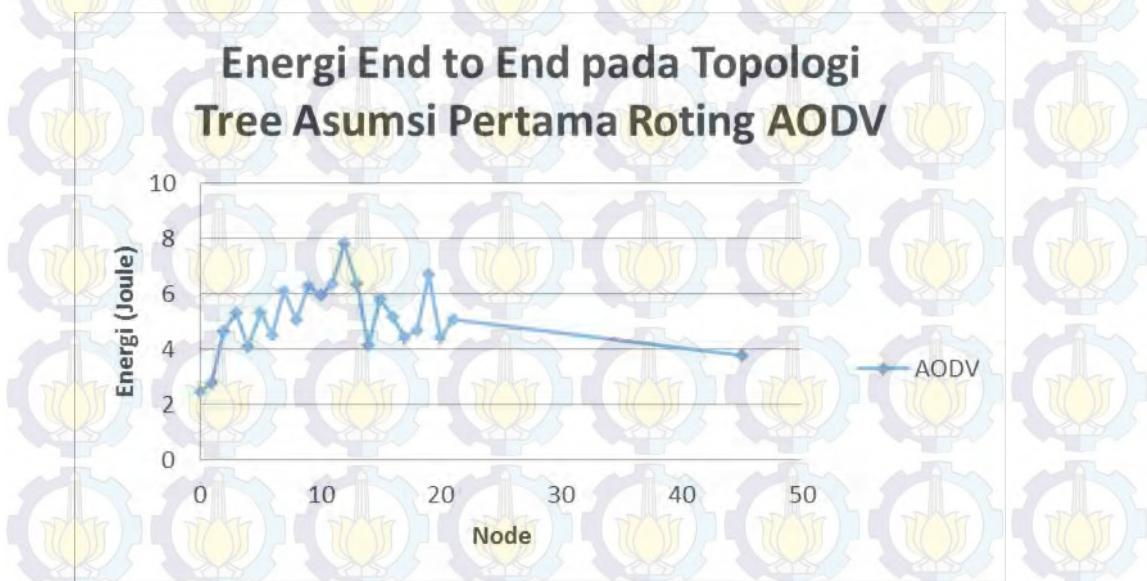
Tabel 4.6. Hasil simulasi topologi tree dengan routing AODV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	7.56 kbps
2	<i>Delay end-to-end</i> rata-rata	19.50 ms
3	<i>Packet Loss</i>	9 paket
4	Energi <i>end-to-end</i> rata-rata	5.01 Joule

Untuk grafik energi end-to-end dapat dilihat pada Gambar 4.10 dan Gambar 4.11 berikut ini.



Gambar 4.10. Energi End-to-end topologi tree asumsi pertama routing DSDV



Gambar 4.11. Energi End-to-end topologi tree asumsi pertama routing AODV

Terlihat pada Gambar 4.10, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,48 Joule, node 1 bernilai 2,79 Joule, node 2 bernilai 4,67 Joule, node 3 bernilai 5,29 Joule, node 4 bernilai 4,12 Joule, node 5 bernilai 5,31 Joule, node 6 bernilai 4,32 Joule, node 7 bernilai 6,1 Joule, node 8 bernilai 5,1 Joule, node 9 bernilai 6,3 Joule, node 10

bernilai 5.96 Joule., node 11 bernilai 6.4 Joule, node 12 bernilai 7.82 Joule, node 13 bernilai 6.4 Joule, node 14 bernilai 4.15 Joule, node 15 bernilai 5.8 Joule dan node 16 bernilai 5.15 Joule, node 17 bernilai 4.41 Joule, node 18 bernilai 4.71 Joule, node 19 bernilai 6.7 Joule, node 20 bernilai 4.41 Joule, node 21 bernilai 5.1 Joule dan node 45 bernilai 3.78 Joule.

Terlihat pada Gambar 4.11, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,48 Joule, node 1 bernilai 2.79 Joule, node 2 bernilai 4.67 Joule, node 3 bernilai 5.29 Joule, node 4 bernilai 4.12 Joule, node 5 bernilai 5.31 Joule, node 6 bernilai 4.32 Joule, node 7 bernilai 6.1 Joule, node 8 bernilai 5.1 Joule, node 9 bernilai 6.3 Joule, node 10 bernilai 5.96 Joule., node 11 bernilai 6.4 Joule, node 12 bernilai 7.82 Joule, node 13 bernilai 6.4 Joule, node 14 bernilai 4.15 Joule, node 15 bernilai 5.8 Joule dan node 16 bernilai 5.15 Joule, node 17 bernilai 4.41 Joule, node 18 bernilai 4.71 Joule, node 19 bernilai 6.7 Joule, node 20 bernilai 4.41 Joule, node 21 bernilai 5.1 Joule dan node 45 bernilai 3.78 Joule.

Berdasarkan Gambar 4.9 dan Gambar 4.10 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 5.01 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 4.74 Joule.

Tabel 4.7 dan Tabel 4.8 merupakan tabel gabungan antara tabel gabungan topologi star, mesh, dan tree dengan asumsi pertama berdasarkan routingsnya.

Tabel 4.7. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi pertama routing AODV

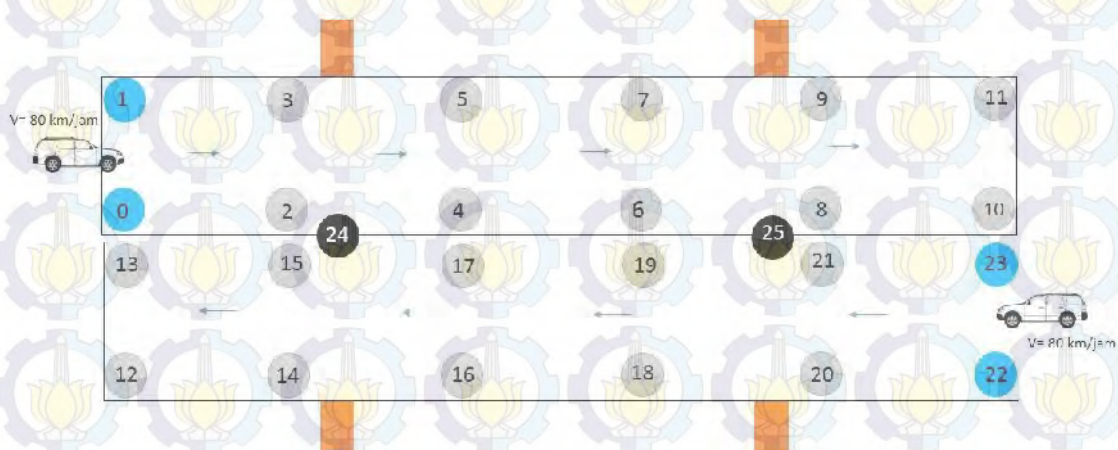
No.	Parameter Kinerja Jaringan	Topologi		
		Star	Mesh	Tree
1	<i>Throughput</i>	0.69 kbps	21.28 kbps	7.56 kbps
2	<i>Delay end-to-end</i> rata-rata	0.532 ms	10.68 ms	19.50 ms
3	<i>Packet Loss</i>	142 paket	1 paket	9 paket
4	Energi <i>end-to-end</i> rata-rata	0.154 Joule	5.66 Joule	5.01 Joule

Tabel 4.8. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi pertama routing DSDV

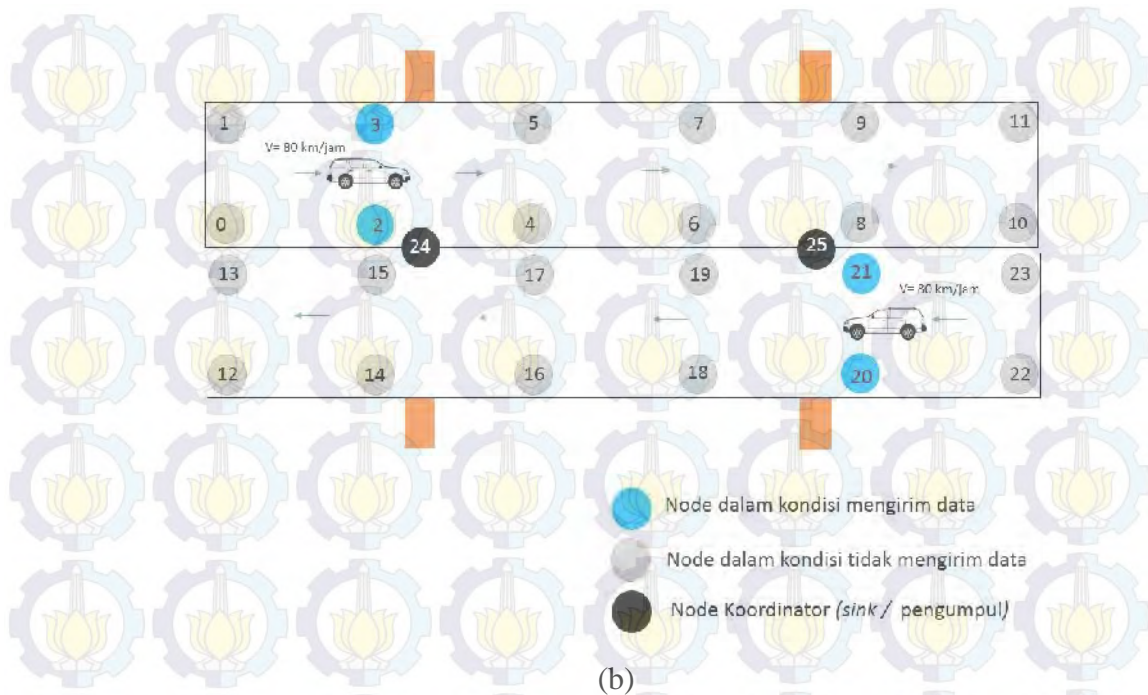
No.	Parameter Kinerja Jaringan	Topologi		
		Star	Mesh	Tree
1	Throughput	0.65 kbps	7.11 kbps	6.35 kbps
2	Delay end-to-end rata-rata	0.840 ms	27,25 ms	26,34 ms
3	Packet Loss	139 paket	2 paket	2 paket
4	Energi end-to-end rata-rata	0.106 Joule	5.09 Joule	4.74 Joule

4.3. Asumsi Kedua

Pada asumsi kedua ini, posisi node di kondisikan jarak antar node terletak dengan jarak 200 meter dan 12 meter. Skenario pada asumsi pertama yakni ketika terdapat sebuah kendaraan yang melintas dengan kecepatan 80 km/jam, sehingga node-node sensor akan mengikuti pergerakan kendaraan dan node yang akan dilewati akan bersiap-siap untuk mengirimkan paket data menuju node koordinator.



(a)

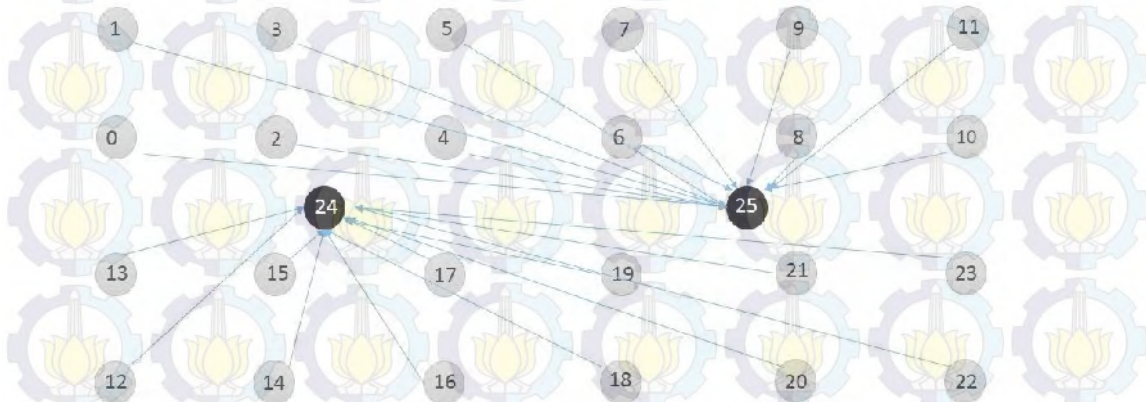


Gambar 4.12 Skenario pada asumsi kedua

Terlihat dari Gambar 4.12 dimana untuk skenario pada asumsi kedua. Pada sisi lajur kiri ke kanan ketika ada kendaraan melaju memasuki jembatan sehingga posisi kendaraan sama dengan node 0 dan node 1 maka node 0 dan node 1 akan mengirim paket data menuju node 25 (node koordinator). Sedangkan pada sisi lajur kanan ke kiri ketika ada kendaraan melaju memasuki jembatan maka node 22 dan node 23 akan mengirim paket data menuju node 24 (node koordinator) seperti pada Gambar 4.12 (a). Node aktif mengirimkan data ditandai dengan node berwarna biru. Pada sisi lajur kiri ke kanan ketika posisi kendaraan sama dengan node 2 dan node 3 maka node 2 dan node 2 akan mengirim paket data menuju node 25, untuk node 0 dan node 1 sudah tidak lagi mengirimkan paket data. Sedangkan Pada sisi lajur kanan ke kiri ketika posisi kendaraan sama dengan node 20 dan node 21 maka node 20 dan node 21 akan mengirim paket data menuju node 24, untuk node 22 dan node 23 sudah tidak lagi mengirimkan paket data seperti pada Gambar 4.12 (b).

4.3.1. Topologi Star

Topologi star merupakan topologi paling sederhana dimana komunikasi terjadi secara *point-to-point* node dengan sink. Desain topologi seperti Gambar 4.11.



Gambar 4.13. Topologi star asumsi pertama

Terlihat pada Gambar 4.13, bahwa pada topologi star setiap node berkomunikasi secara langsung dengan node koordinator (node 24 dan node 25).

Topologi pada asumsi yang ini mirip dengan Gambar 4.13 dimana setiap node mengirim paket data langsung menuju koordinator. Pada sisi kiri ke kanan awalnya, node 0 dan node 1 akan mengirimkan paket data secara bersamaan dan langsung menuju node 25 selama 9 detik, kemudian node 2 dan node 3 akan mengirimkan paket data secara bersamaan dan langsung menuju node 25 selama 9 detik. Dilanjutkan node 4 dan node 5 mengirimkan paket data, lalu node 6 dan node 7, dan yang terakhir adalah node 10 dan node 11 mengirimkan paket data secara bersamaan langsung menuju node 25. Sedangkan Pada sisi kanan ke kiri awalnya, node 22 dan node 23 akan mengirimkan paket data secara bersamaan dan langsung menuju node 24 selama 9 detik, kemudian node 20 dan node 21 akan mengirimkan paket data secara bersamaan dan langsung menuju node 24 selama 9 detik. Dilanjutkan node 18 dan node 19 mengirimkan paket data, lalu node 16 dan node 17, dan yang terakhir adalah node 12 dan node 13 mengirimkan paket data secara bersamaan langsung menuju node 24. Hasil dari simulasi

tampak pada Tabel 4.9. dan Tabel 4.10. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

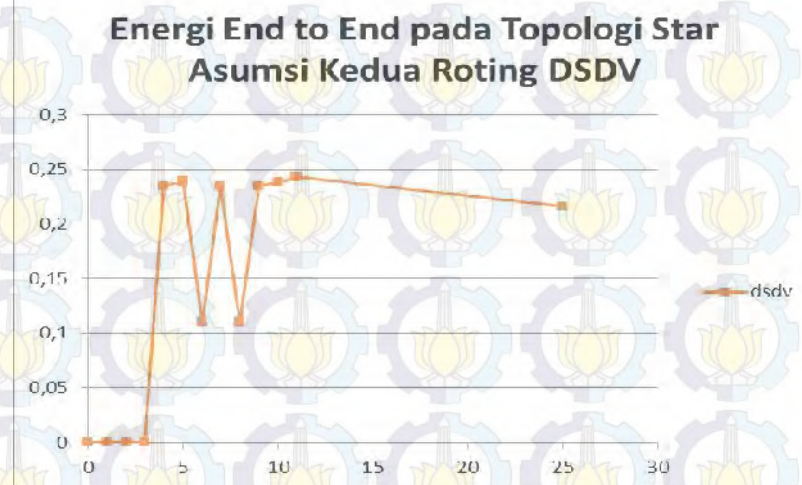
Tabel 4.9. Hasil simulasi topologi star dengan routing DSDV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	0.55 kbps
2	<i>Delay end-to-end</i> rata-rata	0.940 ms
3	<i>Packet Loss</i>	152 paket
4	Energi <i>end-to-end</i> rata-rata	0.143 Joule

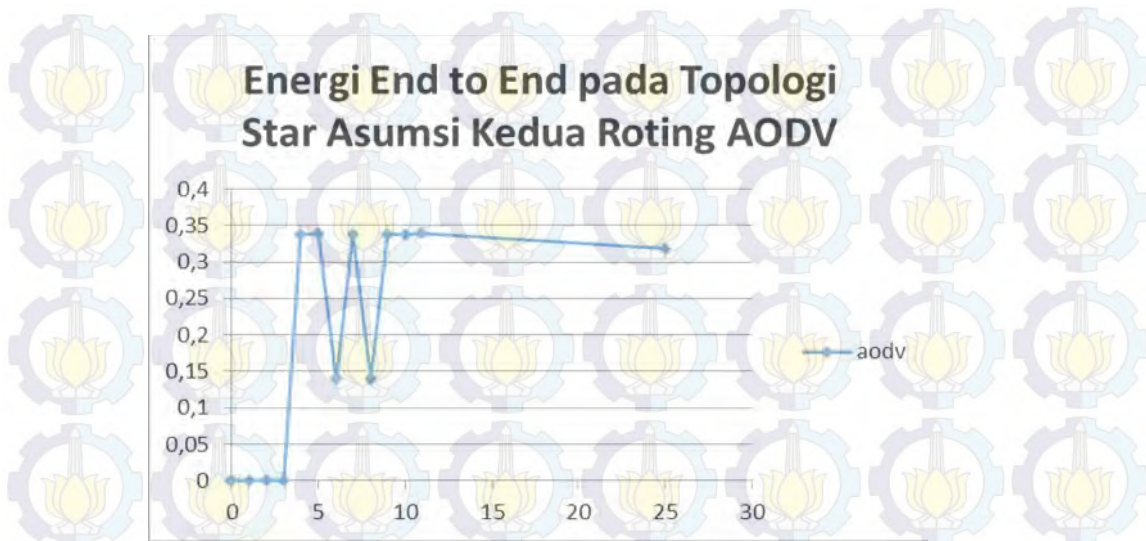
Tabel 4.10. Hasil simulasi topologi star dengan routing AODV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	0.48 kbps
2	<i>Delay end-to-end</i> rata-rata	0.199 ms
3	<i>Packet Loss</i>	149 paket
4	Energi <i>end-to-end</i> rata-rata	0.202 Joule

karena terdapat node yang melebihi jarak jangkauan maka terdapat beberapa paket yang tidak dikirimkan.



Gambar 4.14. Energi End-to-end topologi star asumsi kedua routing DSDV



Gambar 4.15. Energi End-to-end topologi star asumsi kedua routing AODV

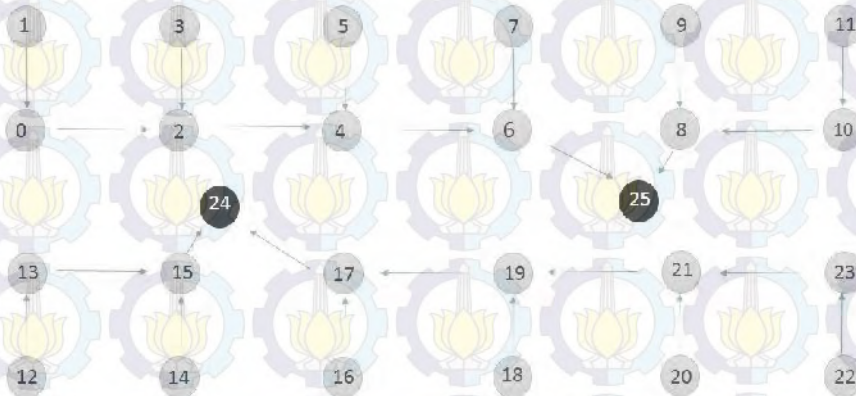
Parameter energi *end-to-end* pada topologi star asumsi pertama untuk routing AODV dan DSDV pada masing-masing node ditunjukkan pada Gambar 4.14 dan Gambar 4.15. Pada lajur kanan ke kiri untuk routing AODV node 0, node 1, node 2 dan node 3, energi *end-to-end* masing-masing node tersebut adalah nol. Hal ini disebabkan ketika node-node tersebut mengirimkan paket data secara langsung menuju node 0, karena jarak yang melebihi jangkauan XBee maka tidak paket tersebut tidak dikirimkan. Sedangkan pada node 4, node 5, node 7, node 8, node 9, node 10 dan node 11 masing-masing memiliki nilai energi *end-to-end* untuk routing AODV berturut-turut 0,338 Joule, 0,339 Joule, 0,338 Joule, 0,338 Joule, 0,338 Joule, dan 0,339 Joule. Node – node ini untuk aktifitas “mendengar”. Node 6 dan node 8 memiliki nilai energi *end-to-end* 0,14 Joule untuk aktifitas pengiriman paket data menuju node 25 (Koordinator). Sedangkan node 25 memiliki nilai energi *end-to-end* 0,318 Joule untuk aktifitas menerima paket data dan “mendengar. Sedangkan untuk routing DSDV berturut-turut 0,235 Joule, 0,239 Joule, 0,235 Joule, 0,235 Joule, 0,238 Joule dan 0,243 Joule. Node 6 dan node 7 memiliki nilai energi *end-to-end* 0,11 untuk aktifitas pengiriman paket data menuju node 25 (Koordinator). Sedangkan node 25 memiliki nilai energi *end-to-end* 0,216 Joule untuk aktifitas menerima paket data dan “mendengar.

Berdasarkan Gambar 4.14 dan Gambar 4.15 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 0,202 Joule.

Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 0,143 Joule.

4.3.2. Topologi Mesh

Topologi Mesh atau biasa disebut komunikasi *peer-to-peer* adalah komunikasi yang terjadi antar node Desain topologi seperti Gambar 4.16.



Gambar 4.16. Topologi mesh asumsi kedua

Terlihat pada Gambar 4.16 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya. Pada skenario dengan asumsi pertama ini, pada sisi kiri ke kanan, ketika sebuah mobil memiliki koordinat yang sama dengan node 1 dan node 0, maka node 1 mengirimkan data melalui node 0 yang kemudian di teruskan dari node 0 menuju node 2. Node 2 juga menerima data dari node 3 dan seterusnya sehingga data sampai di node 25. Sedangkan pada sisi kanan ke kiri, ketika sebuah mobil memiliki koordinat yang sama dengan node 22 dan node 23, maka node 22 mengirimkan data melalui node 23 yang kemudian di teruskan dari node 23 menuju node 21. Node 21 juga menerima data dari node 20 dan seterusnya sehingga data sampai di node 24.

Hasil dari simulasi tampak pada Tabel 4.11. dan Tabel 4.12. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

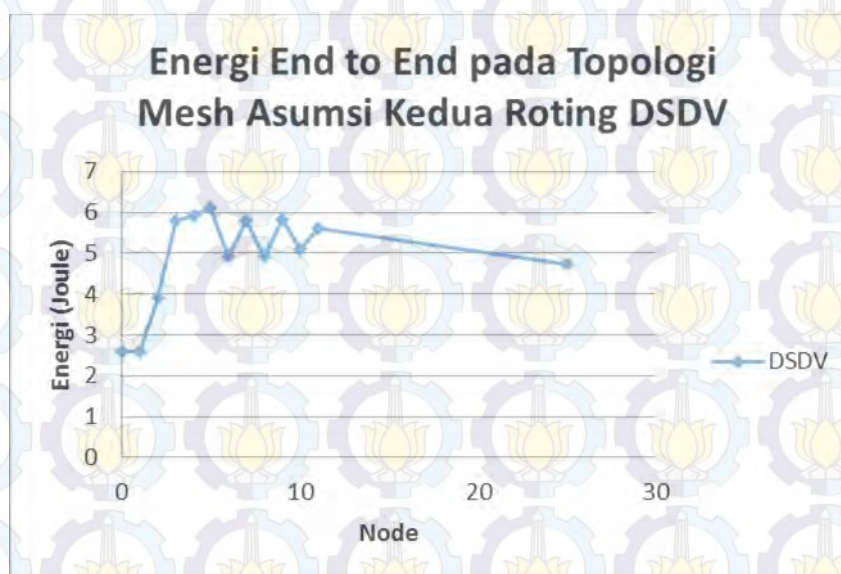
Tabel 4.11. Hasil simulasi topologi mesh dengan routing DSDV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	5.35 kbps
2	<i>Delay end-to-end</i> rata-rata	27.45 ms
3	<i>Packet Loss</i>	4 paket
4	Energi <i>end-to-end</i> rata-rata	4.91 Joule

Tabel 4.12. Hasil simulasi topologi mesh dengan routing AODV dengan asumsi kedua

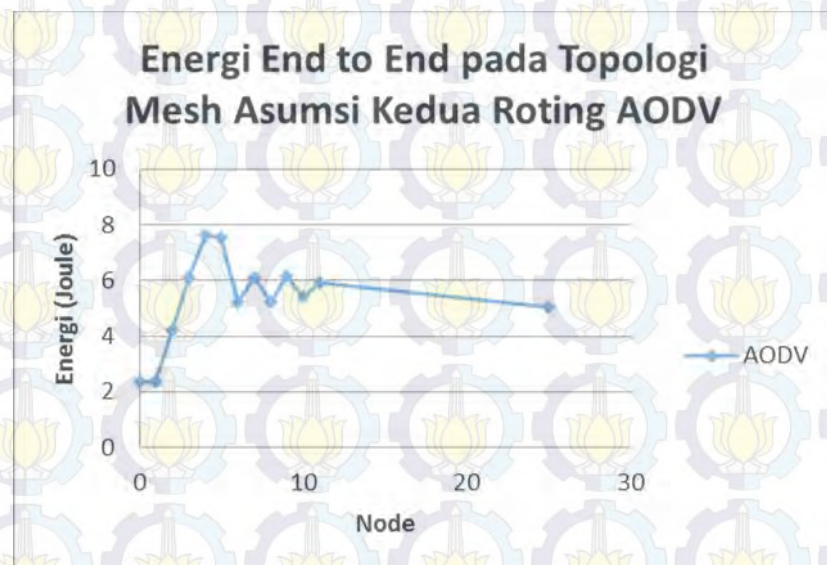
No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	19.89 kbps
2	<i>Delay end-to-end</i> rata-rata	19.65 ms
3	<i>Packet Loss</i>	2 paket
4	Energi <i>end-to-end</i> rata-rata	5.32 Joule

Untuk grafik energi end-to-end dapat dilihat pada Gambar 4.17 dan Gambar 4.18 berikut ini.



Gambar 4.17. Energi End-to-end topologi mesh asumsi kedua routing DSDV

Terlihat pada Gambar 4.17, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2.58 Joule, node 1 bernilai 2.58 Joule, node 2 bernilai 3.91 Joule, node 3 bernilai 5.8 Joule, node 4 bernilai 5.93 Joule, node 5 bernilai 6.12 Joule, node 6 bernilai 4.93 Joule, node 7 bernilai 4.8 Joule, node 8 bernilai 5.93 Joule, node 9 bernilai 5.83 Joule, node 10 bernilai 5.1 Joule dan node 11 bernilai 5.6 Joule, sedangkan node 25 bernilai 4.72 Joule.



Gambar 4.18. Energi End-to-end topologi mesh asumsi kedua routing AODV

Terlihat pada Gambar 4.18, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2.38 Joule, node 1 bernilai 2.38 Joule, node 2 bernilai 4.31 Joule, node 3 bernilai 6.1 Joule, node 4 bernilai 7.63 Joule, node 5 bernilai 7.52 Joule, node 6 bernilai 5.23 Joule, node 7 bernilai 6.1 Joule, node 8 bernilai 5.23 Joule, node 9 bernilai 6.13 Joule, node 10 bernilai 5.4 Joule dan node 11 bernilai 5.9 Joule, sedangkan node 25 bernilai 5.02 Joule.

Berdasarkan Gambar 4.17 dan Gambar 4.18 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 5.32 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 4.91 Joule.

4.3.3. Topologi Tree

Desain topologi tree dapat dilihat pada Gambar 4.19 berikut ini.



Gambar 4.19. Topologi tree asumsi kedua

Terlihat pada Gambar 4.19 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya. Pada topologi ini hampir sama dengan topologi mesh, namun pada topologi ini komunikasi node terbagi atas beberapa jalur untuk sampai ke koordinator.

Tabel 4.13. Hasil simulasi topologi tree dengan routing DSDV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	5.71 kbps
2	<i>Delay end-to-end</i> rata-rata	25.97 ms
3	<i>Packet Loss</i>	5 paket
4	Energi <i>end-to-end</i> rata-rata	4.64 Joule

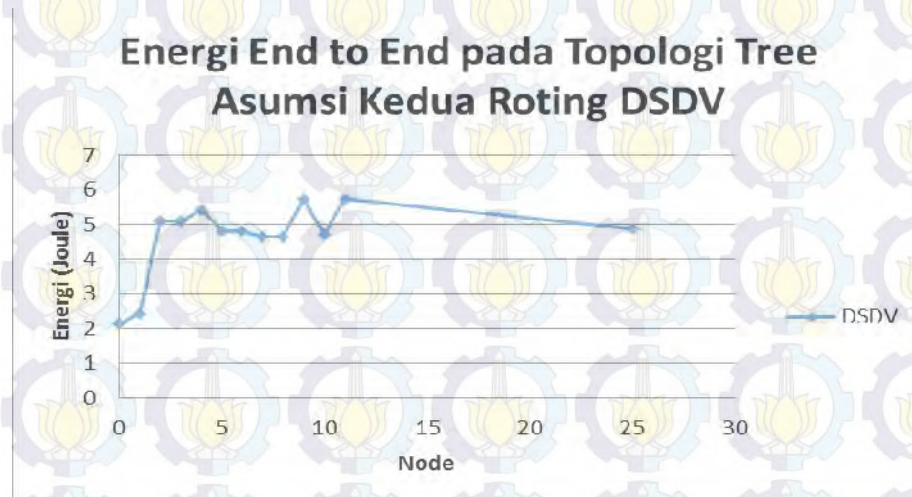
Untuk sisi kiri ke kanan, jalur pertama yakni dari node 1 menuju node 3, node 5, node 7, node 6 baru ke node 25. Jalur kedua yakni node 0 menuju node 2, node 4, node 6 baru ke node 25. Jalur ke tiga dari node 9 ke node 11, node 10, node 8 baru ke node 25. Sedangkan sisi kanan ke kiri, jalur pertama yakni dari

node 23 menuju node 21, node 19, node 17 baru ke node 24. Jalur kedua yakni node 22 menuju node 20, node 18, node 16 baru ke node 24. Jalur ke tiga dari node 14 ke node 12, node 13, node 15 baru ke node 24. Hasil dari simulasi tampak pada Tabel 4.13. dan Tabel 4.14. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

Tabel 4.14. Hasil simulasi topologi tree dengan routing AODV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	6.98 kbps
2	<i>Delay end-to-end</i> rata-rata	20.43 ms
3	<i>Packet Loss</i>	8 paket
4	Energi <i>end-to-end</i> rata-rata	4.52 Joule

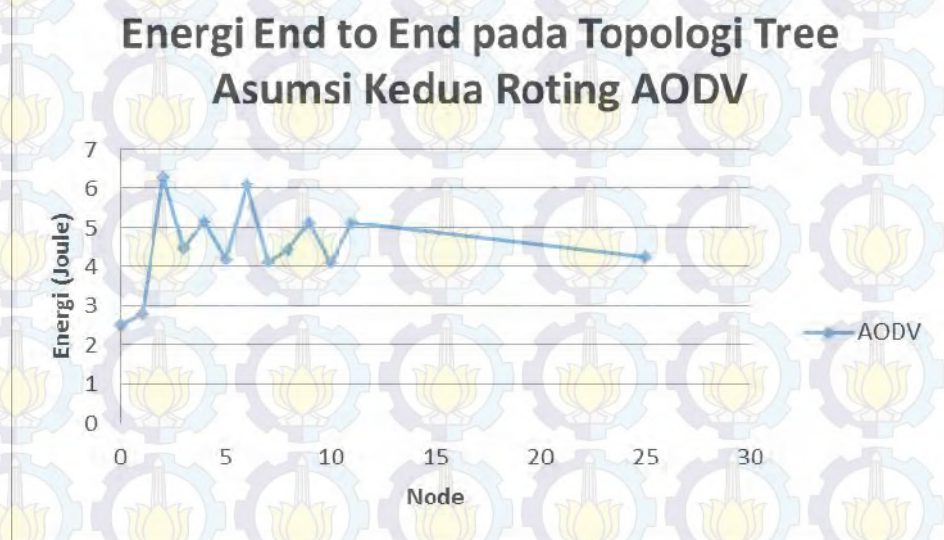
Untuk grafik energi end-to-end dapat dilihat pada Gambar 4.20 dan Gambar 4.21 berikut ini.



Gambar 4.20. Energi End-to-end topologi tree asumsi kedua routing DSDV

Terlihat pada Gambar 4.20, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,12 Joule, node 1

bernilai 2.41 Joule, node 2 bernilai 5.1 Joule, node 3 bernilai 5.08 Joule, node 4 bernilai 5.4 Joule, node 5 bernilai 4.81 Joule, node 6 bernilai 4.81 Joule, node 7 bernilai 4.65 Joule, node 8 bernilai 4.65 Joule, node 9 bernilai 5.73 Joule, node 10 bernilai 4.7 Joule., node 11 bernilai 5.73 Joule. Sedangkan node 25 bernilai 4.67 Joule.



Gambar 4.21. Energi End-to-end topologi tree asumsi kedua routing AODV

Terlihat pada Gambar 4.21, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,52 Joule, node 1 bernilai 2.81 Joule, node 2 bernilai 6.3 Joule, node 3 bernilai 4.48 Joule, node 4 bernilai 5.15 Joule, node 5 bernilai 4.2 Joule, node 6 bernilai 6.1Joule, node 7 bernilai 4.15 Joule, node 8 bernilai 4.45 Joule, node 9 bernilai 5.13 Joule, node 10 bernilai 4.1 Joule., node 11 bernilai 5.13 Joule. Sedangkan node 25 bernilai 4.27 Joule.

Berdasarkan Gambar 4.20 dan Gambar 4.21 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 4.523 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 4.64 Joule

Tabel 4.15 dan Tabel 4.16 merupakan tabel gabungan antara tabel gabungan topologi star, mesh, dan tree dengan asumsi kedua berdasarkan routingnya.

Tabel 4.15. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi kedua routing AODV

No.	Parameter Kinerja Jaringan	Topologi		
		Star	Mesh	Tree
1)	<i>Throughput</i>	0.48 kbps	19.89 kbps	6.98 kbps
2)	<i>Delay end-to-end</i> rata-rata	0.199 ms	19.65 ms	20.43 ms
3)	<i>Packet Loss</i>	149 paket	2 paket	8 paket
4)	Energi <i>end-to-end</i> rata-rata	0.202 Joule	5.32 Joule	4.52 Joule

Tabel 4.16. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi kedua routing DSDV

No.	Parameter Kinerja Jaringan	Topologi		
		Star	Mesh	Tree
1	<i>Throughput</i>	0.55 kbps	5.35 kbps	6.98 kbps
2	<i>Delay end-to-end</i> rata-rata	0.940 ms	27.45 ms	20.43 ms
3	<i>Packet Loss</i>	152 paket	4 paket	8 paket
4	Energi <i>end-to-end</i> rata-rata	0.143 Joule	4.91 Joule	4.52 Joule

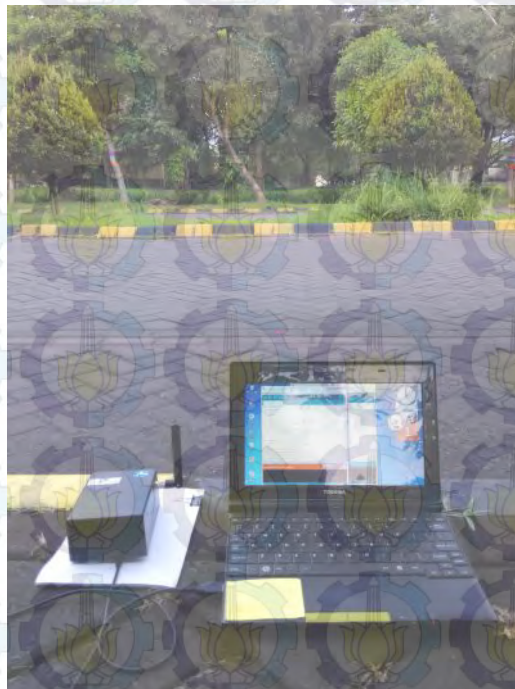
4.4. Analisa

Dari hasil yang didapat terlihat bahwa antara asumsi pertama memiliki nilai-nilai kinerja yang baik khususnya pada topologi mesh dengan routing AODV namun pada asumsi ini memerlukan banyak node. Sedangkan pada asumsi kedua nilai-nilai kinerja cukup baik. Sebanding dengan jarak yang semakin jauh maka nilai *throughput* semakin kecil. Pada asumsi ini memiliki kelebihan node

yang sedikit sehingga mampu meminimalisir jumlah node, sehingga mampu lebih meminimalisir energi yang diperlukan.

4.5. Validasi Parameter Simulasi

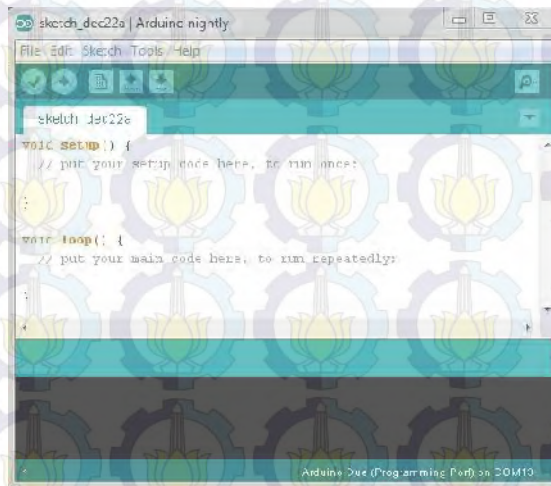
Pada proses validasi parameter simulasi ini dilakukan pengujian untuk parameter yang digunakan dalam simulasi. Pengujian ini dengan mengirimkan data sensor accelerometer kemudian melihat apakah data tersebut dapat diterima oleh node koordinator/*sink* dengan baik atau tidak. Selanjutnya juga menghitung *packet loss*, *delay*, *throughput* dan konsumsi arus yang digunakan dari baterai. Setelah menetapkan terlebih dahulu posisi node yang di bagi dalam dua asumsi yakni asumsi pertama jarak antara koordinator ke sensor 1 yakni 12 meter sama dengan jarak sensor 3 ke sensor 2, dan sensor 2 ke koordinator 100 meter sama dengan jarak sensor 3 ke sensor 1. Sedangkan asumsi kedua jarak antara koordinator ke sensor 1 yakni 12 meter sama dengan jarak sensor 3 ke sensor 2, sedangkan sensor 2 ke koordinator 200 meter sama dengan jarak sensor 3 ke sensor 1.



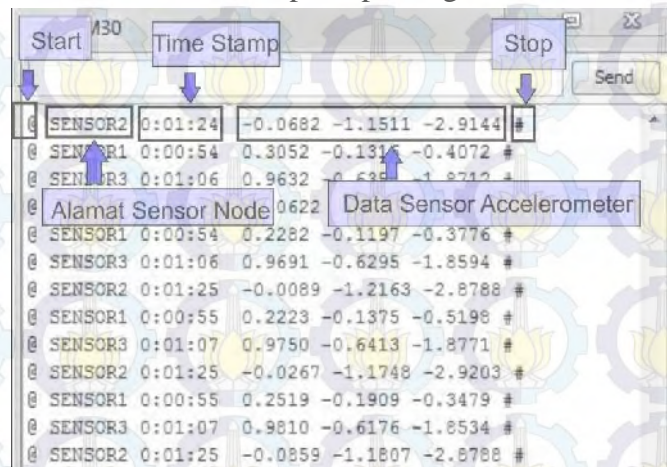
Gambar 4.22. Proses pengambilan data secara real

4.6. Pengujian Kalibrasi Node

Pengujian dilakukan pada node koordinator. Pengujian dilakukan untuk mendapatkan data pembacaan sensor accelerometer sebelum dikirimkan pada node koordinator. Berikut Gambar 4.24 menunjukkan data yang diterima oleh penerima melalui port serial yang terhubung dengan desktop ataupun notebook.



Gambar 4.23 Tampilan pemrograman arduino



Gambar 4.24 window serial monitoring pada node koordinator

4.7. Manajemen Transmisi Data

Proses manajemen transmisi data pada penelitian ini model komunikasi berdasarkan terjadinya suatu kejadian. Node yang telah dibagi dalam 2 macam yakni node sensor dan node koordinator/sink. Node sensor memiliki fungsi sebagai sensing node yang akan mensensing adanya getaran yang tertangkap di daerah sensing node. Node sensor yang berada paling ujung (*End device*) melanjutkan pengiriman datanya melalui node sensor terdekatnya. Node sensor

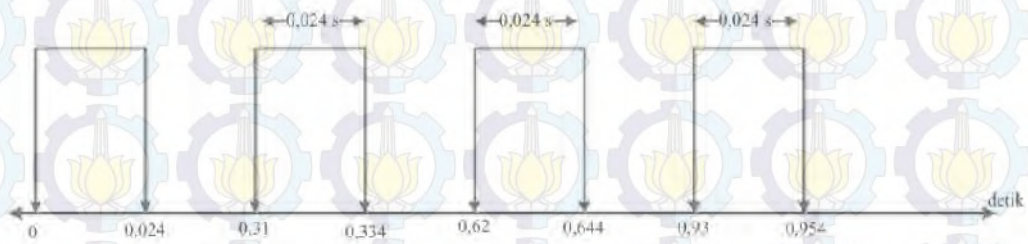
yang berada dekat dengan node koordinator (router) akan menerima data dari node sensor sebelumnya dan data hasil pembacaan sensor dikirimkan pada node koordinator. Sedangkan node koordinator berfungsi sebagai pengumpul data sensing dari node sensor yang kemudian data akan dikirimkan pada base station. sensor sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *awake* (node aktif), Sedangkan ketika node sensor tidak sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *sleep*. *Sleep* bukan berarti node mati namun node berada pada kondisi *idle* atau hanya mendengar saja. Pada tahap penginderaan yang dilakukan pada penelitian ini adalah kalibrasi dari sensor memastikan sensor yang digunakan dapat berfungsi dengan baik. Berikut pengujian dari proses transmisi data sensor accelerometer yang digunakan pada node jaringan sensor di sistem monitoring kesehatan struktur ini. Dari penelitian yang dilakukan, desain terbaik terdapat pada topologi mesh dengan desain seperti pada Gambar 4.14. Disini ketika node 1 akan mengirim data, melewati node 2 kemudian dari node 2 mengirimkan data ke node koordinator. Kemudian dilanjutkan ketika node 0 akan mengirim data ke node 2 kemudian dari node 2 akan mengirim ke node koordinator. Penentuan node selanjutnya berdasarkan node terdekat dengan jarak terdekat berdasarkan route terdekat dan jumlah energi dari setiap node. Ketika node akan berlanjut ke node selanjutnya node melakukan pengecekan terhadap energinya ketika energi cukup maka proses dilanjutkan. Begitu seterusnya sampai semua node terlewati. Pada topologi jaringan mesh dicover oleh dua buah sink node 24 dan 25. Cluster pertama dicover oleh node 24 dengan node 0,1,2,3,4,5,12,13,14,15,16 dan 17. Kemudian cluster kedua dicover oleh node 25 dengan node 6,7,8,9,10,11,18,19,20,21,22 dan 23. Jadi setiap koordinator mengcover 12 node dan 1 node koordinator.

Sedangkan dalam proses pentransmisi data besar byte yang dikirim adalah 12 byte dengan perincian sesuai Gambar 4.25 berikut :

Start byte (1 byte)	Alamat Sensor dan Jenis Sensor (2 byte)	Data Monitoring (8 byte)	Stop byte (1 byte)
------------------------	--	-----------------------------	-----------------------

Gambar 4.25 Format Protokol Pemaketan Data pada Node Sensor

Pada proses pengiriman data setiap paket berisi 12 Byte. Sehingga ketika pengiriman data dengan panjang byte 1000 byte terkirim dalam jangka 2 detik. jadi dapat disimpulkan bahwa setiap 1 paket dengan panjang 12 byte terkirim dalam jangka waktu 0,024 detik. Ilustrasi pengiriman dapat dilihat pada Gambar 4.26.



Gambar 4.26 Waktu Pengiriman data

Dari hasil yang diujikan dapat disimpulkan bahwa node sensor dapat menerima pengindraan getaran dari kendaraan yang melintas dan mengirimkan langsung secara singlehop dan dihitung berapakah konsumsi arus dari satu node sensor ini selama satu jam pengujian. Pada kondisi LOS node sensor dicatu oleh baterai. Dengan kapasitas 220 mA dalam waktu satu jam pengukuran. Pengukuran berlangsung ketika siang hari kondisi jalanan dengan intensitas kendaraan cukup ramai. Pada awal pengukuran baterai diukur dahulu dalam kondisi close circuit voltage dengan multimeter sebesar 9,6 Volt. Setelah satu jam pengukuran, diukur kembali pada baterai untuk mengetahui berapakah yang dibutuhkan oleh node sensor satu jam pengukuran dalam kondisi trafik jalanan yang cukup padat. Kemudian besar tegangan baterai sebesar 8,5 Volt ini saat kondisi node *awake*, sedangkan pada kondisi node *sleep (idle)* yakni kondisi dimana sedang tidak ada kendaraan yang lewat, besar tegangan setelah 1 jam yakni 8,7 Volt. Jadi dapat disimpulkan bahwa satu jam tegangan yang digunakan saat kondisi node *awake* (aktif) sebesar 1,1 Volt, sedangkan node *sleep (idle)* sebesar 0.9 Volt. Perhitungan besar konsumsi arus yang diperlukan pada kondisi node *awake* (aktif) dan node *sleep (idle)* berdasarkan rumus berikut ini :

$$\frac{\text{Besar Tegangan Terukur}}{\text{Isi Baterai}} \times 100 \times 0.22Ah = 4.1$$

Karena node dicatu oleh baterai dengan *brand* Imedion berkapasitas 9.6 V dengan kapasitas arus 220 mA atau sama dengan 0.22 Ah. Maka untuk Isi baterai sebesar 9.6 V. Kemudian data dimasukkan pada rumus diatas untuk mendapat konsumsi arus selama satu jam. Berikut perhitungannya:

- Kondisi node *awake* (aktif)

$$\frac{1,1}{9,6} \times 100 \times 0.22 \text{Ah} = 2,52 \text{Ah}$$

- Kondisi node *sleep* (*idle*)

$$\frac{0,9}{9,6} \times 100 \times 0.22 \text{Ah} = 2,06 \text{Ah}$$

Didapatkan kondisi bahwa pada saat node *awake* (aktif) membutuhkan konsumsi arus sebesar 2.52 Ah sedangkan saat kondisi node *sleep* (*idle*) membutuhkan konsumsi arus sebesar 2.06 Ah selama satu jam pengukuran. Berikut pada Tabel 4.17 menunjukkan perbandingan konsumsi arus dari kedua kondisi node.

Tabel 4.17 Perbandingan Konsumsi Arus Berdasarkan Kondisi Node

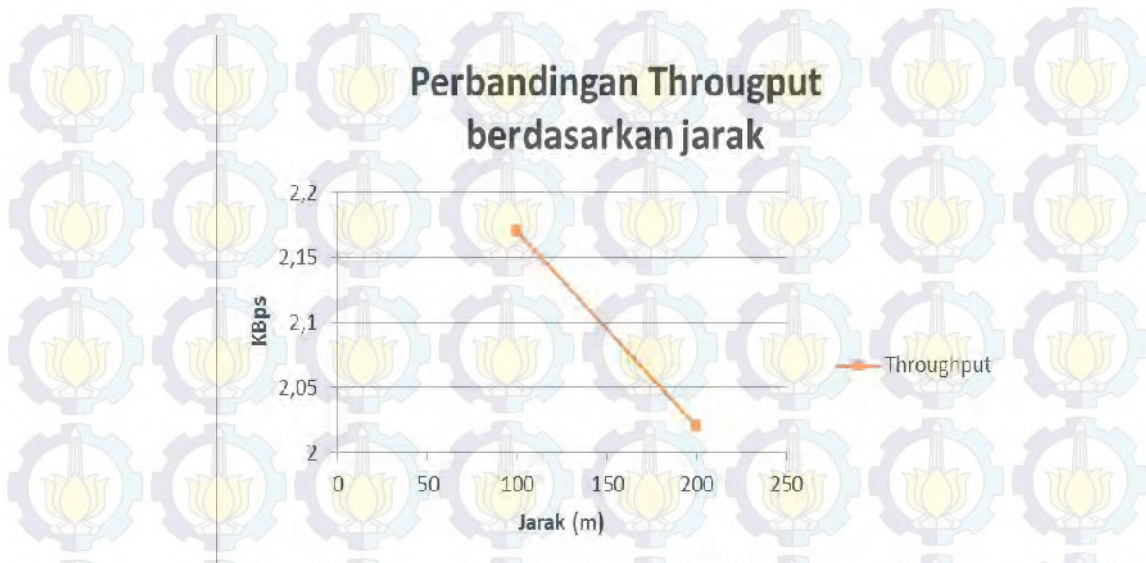
Kondisi Node	Tegangan (V)	Konsumsi Arus (Ah)
Sleep	0.9	2.06
Awake	1.1	2.52

4.8. Analisa Validasi Parameter Simulasi

Pada proses validasi parameter simulasi ini didapatkan hasil perbandingan antara jarak 100 meter dengan jarak 200 meter. Didapatkan hasil seperti pada Tabel 4.18 berikut ini :

Tabel 4.18 Hasil Pengukuran *Throughput*

Jarak	Paket data yang diterima dalam 1 menit	<i>Throughput</i> node (KBps)
	node (byte)	
100 m	130300	2.17
200 m	121100	2.02



Gambar 4.27 Perbandingan throughput validasi

Pada pengukuran *throughput* Tabel 4.13 dengan jarak 100 meter pada node sensor, node koordinator menerima data paket sebesar 130300 Byte dari node dengan persamaan 3.6 diperoleh nilai *throughput* sebesar 2.17 KBps, sedangkan untuk jarak 200 m pada node sensor, node koordinator menerima data paket sebesar 121100 Byte dari node diperoleh nilai *throughput* sebesar 2.02 KBps. Terlihat pada Tabel 4.13 mengalami penurunan sekitar 0.15 KBps. Hal ini disebabkan banyaknya paket data yang hilang karena pengaruh propagasi, dengan jarak yang semakin jauh membutuhkan waktu propagasi yang besar/*delay*, sehingga paket yang diterima dalam 1 menit berkurang.

Sementara itu dalam Pengujian *delay* disini dilakukan untuk mengetahui kinerja sistem terhadap pengaruh jarak terhadap lama waktu pengiriman (end to end delay). Pada pengujian delay ini peneliti mengirimkan data sebesar 100 byte, kemudian dalam pengiriman data tersebut jarak diubah berdasarkan parameter simulasi, hal ini untuk mengetahui pengaruh jarak terhadap lama pengiriman dengan melihat time stamp yang ditunjukkan pada saat penerimaan data di node koordinator. Pengukuran delay yang terjadi pada node sensor dapat dilihat pada Tabel 4.19. Gambar 4.25 menunjukkan grafik pengaruh jarak terhadap delay penerimaan data pada node koordinator.

Tabel 4.19 Hasil Pengukuran Delay

Jarak	Delay node (ms)
100 m	39
200 m	67



Gambar 4.28 Perbandingan delay validasi

Untuk konsumsi arus yang digunakan pengamatan dilakukan dengan dua macam rancangan dimana rancangan pertama konsumsi arus di hitung dengan kondisi node tanpa bebas dan rancangan kedua kondisi node dengan beban sensor. Pada setiap node dicatu oleh baterai dengan *brand* Imedion berkapasitas 9.6 V. Pada kondisi baterai terisi penuh kapasitas arus 220 mA kemudian ketika diukur arusnya setiap 30 menit maka di dapatkan hasil sebagai berikut.

Tabel 4.20 Hasil pengukuran konsumsi arus

Kondisi	Sisa kapasitas saat pengukuran 30 menit pertama	Konsumsi rata-rata tiap 30 menit	Waktu habis
Tanpa Beban	207.1 mA	13 mA	8 Jam
Dengan Beban	194.3 mA	26 mA	4 Jam

Terlihat dari Tabel 4.20 bahwa untuk kondisi tanpa beban pada 30 menit pertama sisa kapasitas arus sebesar 207.1 mA sehingga rata-rata konsumsi arus tiap 30 menit akan berkurang sebanyak 13 mA maka baterai akan habis dalam

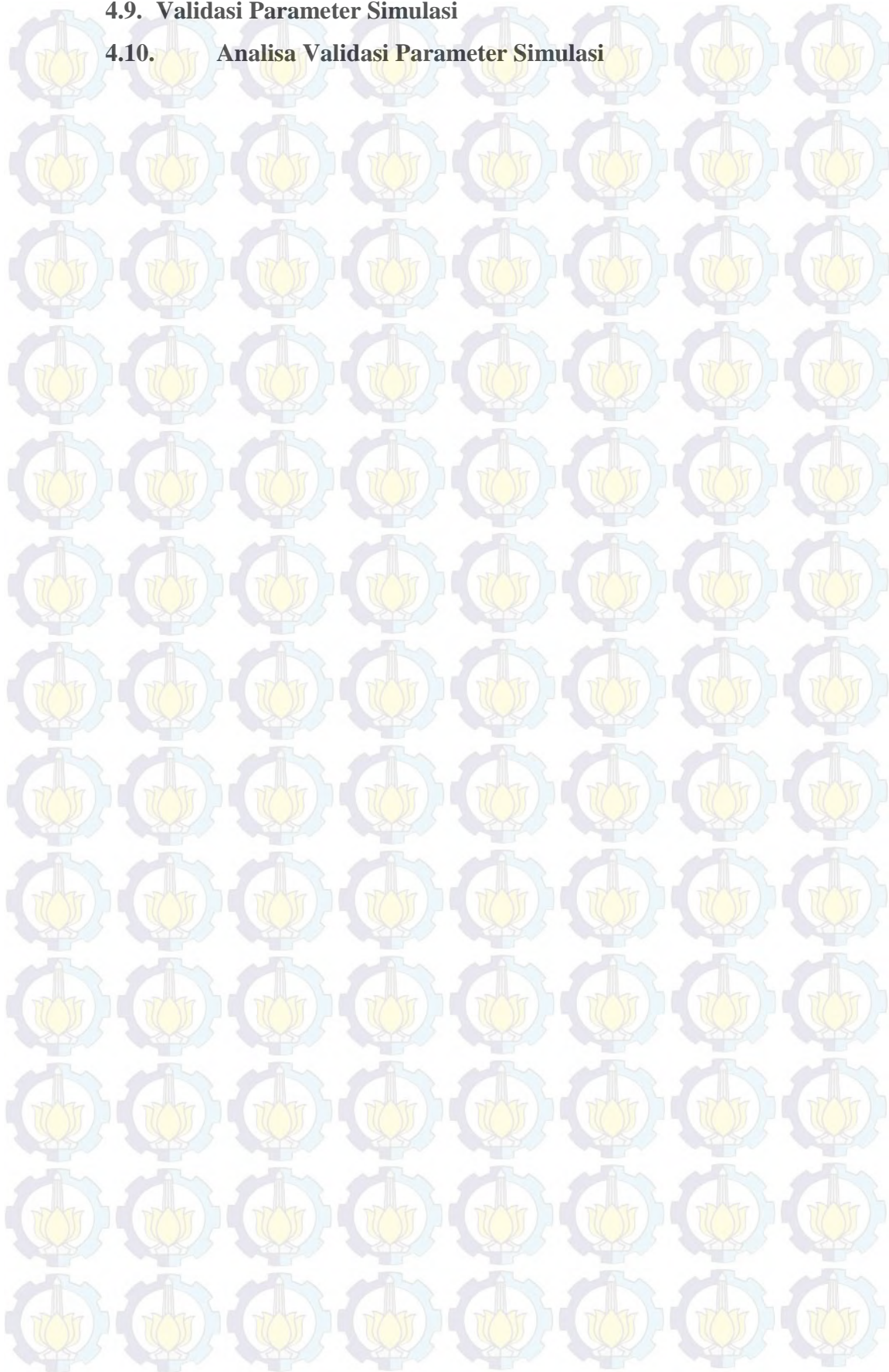
waktu 8 jam sedangkan pada kondisi dengan beban pada 30 menit pertama sisa kapasitas arus sebesar 194.3 mA sehingga rata-rata konsumsi arus tiap 30 menit akan berkurang sebanyak 26 mA maka baterai akan habis dalam waktu 4 jam.



(HALAMAN INI SENGAJA DIKOSONGKAN)

4.9. Validasi Parameter Simulasi

4.10. Analisa Validasi Parameter Simulasi



BAB 5 KESIMPULAN DAN SARAN

5.1. KESIMPULAN

Pada sub bab ini, akan dijelaskan mengenai kesimpulan-kesimpulan yang diambil berdasarkan hasil simulasi dan analisa dari bab 4 dengan metode bab 3. Kesimpulan yang dapat diambil adalah :

- 1) Dari hasil analisa proses manajemen energi pada sistem monitoring struktur jembatan dapat dilakukan dengan memilih topologi yang terbaik dengan routing yang baikpula.
- 2) Asumsi pertama adalah skenario dimana jarak antar node sebesar 100 meter x 12 meter. Didapatkan hasil sebagai berikut :
 - a. Topologi *star* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 139 packet *delay* sebesar 0.840 ms, *throughput* sebesar 0.65 kbps dan energi sebesar 0.106 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 142 packet, *delay* sebesar 0.532 ms, *throughput* sebesar 0,69 kbps dan energi sebesar 0.145 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *star* tidak dapat diimplementasikan pada sistem SMKS jembatan.
 - b. Topologi *mesh* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 2 packet, *delay* sebesar 27.25 ms, *throughput* sebesar 7.11 kbps dan energi sebesar 5.62 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 1 packet, *delay* sebesar 10.68 ms, *throughput* sebesar 21.28 kbps dan energi sebesar 5.99 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan.
 - c. Topologi *tree* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 2 packet, *delay*

sebesar 26.34 ms, *throughput* sebesar 6.35 kbps dan energi sebesar 5.19 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 9 *packet*, *delay* sebesar 19.50 ms, *throughput* sebesar 7.56 kbps dan energi sebesar 5.66 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan.

3) Asumsi kedua adalah skenario dimana jarak antar node sebesar 200 meter x 12 meter. Didapatkan hasil sebagai berikut :

- a. Topologi *star* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 152 *packet*, *delay* sebesar 0.940 ms, *throughput* sebesar 0.55 kbps dan energi sebesar 0.143 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 149 *packet*, *delay* sebesar 0.652 ms, *throughput* sebesar 0.48 kbps dan energi sebesar 0.199 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *star* tidak dapat diimplementasikan pada sistem SMKS jembatan.
- b. Topologi *mesh* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 4 *packet*, *delay* sebesar 27.45 ms, *throughput* sebesar 5.35 kbps dan energi sebesar 5.10 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 2 *packet*, *delay* sebesar 19.65 ms, *throughput* sebesar 19.89 kbps dan energi sebesar 5.31 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan.
- c. Topologi *tree* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 5 *packet*, *delay* sebesar 25.97 ms, *throughput* sebesar 5.71 kbps dan energi sebesar 54.60 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 8

packet, *delay* sebesar 20.43 ms, *throughput* sebesar 6.98 kbps dan energi sebesar 4.46 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan. Topologi yang dapat diterapkan pada SHM jembatan adalah topologi *mesh* dan *tree*.

- 4) Topologi yang terbaik dan dapat diterapkan pada Sistem SHM Jembatan pada penelitian ini adalah topologi *mesh*.
- 5) Pada asumsi pertama, penerapan topologi *mesh* dengan routing AODV lebih unggul dibandingkan topologi *star* dan *tree*.
- 6) Pada asumsi kedua, penerapan topologi *mesh* dengan routing AODV lebih unggul dibandingkan dengan topologi *star* dan *tree*.
- 7) Validasi yang dilakukan adalah validasi parameter simulasi dengan jarak 100 meter x 12 meter dan 200 meter x 12 meter sesuai dengan asumsi yang digunakan.

5.2. SARAN

Untuk kemajuan penelitian selanjutnya, terdapat beberapa saran setelah penelitian ini dilakukan. Saran yang dapat disampaikan adalah :

- 1) Untuk penelitian selanjutnya dapat menggunakan karakteristik sensor-sensor yang lain seperti sensor *strain gauge*, sensor suhu, dan *wind sensor*.
- 2) Pada penelitian ini hanya berbasis simulasi sehingga penelitian selanjutnya dapat menggunakan perangkat real dan dengan menguji topologi dan routing penelitian ini.
- 3) Untuk data real pada penelitian ini hanya sebagai validasi parameter dalam simulasi sehingga pada penelitian selanjutnya dapat menghitung data secara real sesuai hasil simulasi.



LAMPIRAN A

```
#
=====
# Define options
#
=====
set val(chan) Channel/WirelessChannel ;# Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-
propagation model
set val(netif) Phy/WirelessPhy/802_15_4
set val(mac) Mac/802_15_4
set val(ifq) Queue/DropTail/PriQueue ;# interface
queue type
set val(ll) LL ;# link layer
type
set val(ant) Antenna/OmniAntenna ;# antenna
model
set val(ifqlen) 150 ;# max packet
in ifq
set val(nn) 26 ;# number of
mobilenodes
set val(rp) AODV ;# routing
protocol
set val(x) 1500
set val(y) 50
# =====
# =====
# Antenna Settings
# =====
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 0.6682
Antenna/OmniAntenna set Gt_ 2.1
Antenna/OmniAntenna set Gr_ 2.1
```

```

#####
#Physical layer setting
#####
Phy/WirelessPhy set freq_ 2.4e+9 ;# The working band is 2.4GHz
Phy/WirelessPhy set L_ 1.0 ;#Define the system loss in
TwoRayGround
Phy/WirelessPhy set pt_ 0.01
Phy/WirelessPhy set RXThresh_ 1.0e-13

# Initialize Global Variables
set ns_ [new Simulator]

# Tell teh simulator to use teh new type trace data
set tesis200 [open tesis200.tr w]
$ns_ trace-all $tesis200

#Define the NAM output file
set tesis200 [open tesis200.nam w]
$ns_ namtrace-all-wireless $tesis200 $val(x) $val(y)

# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

# Create God
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]

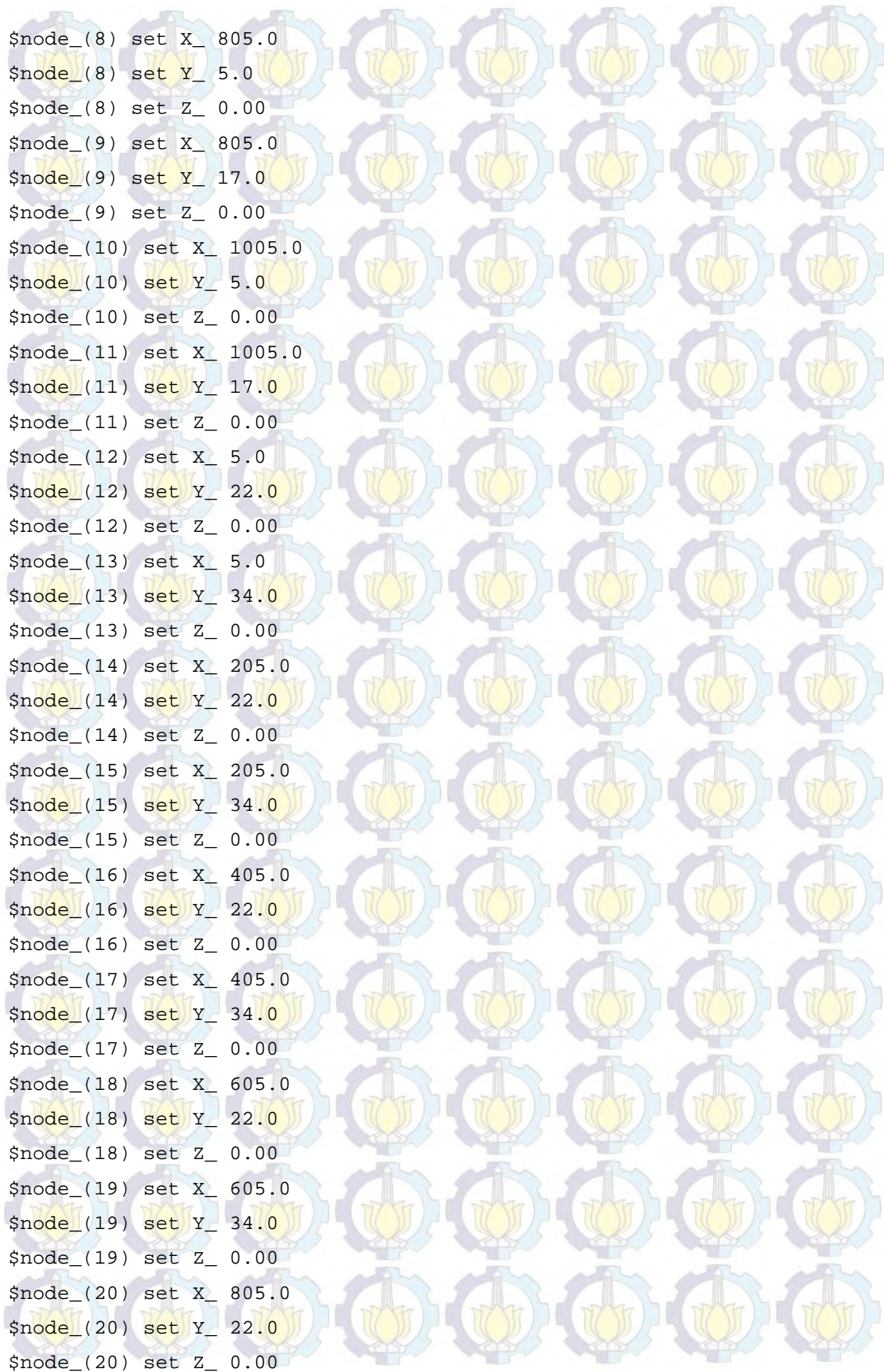
# configure node
$ns_ node-config -adhocRouting $val(rp) \
  -llType $val(ll) \
  -macType $val(mac) \
  -ifqType $val(ifq) \
  -ifqLen $val(ifqlen) \
  -antType $val(ant) \
  -propType $val(prop) \
  -phyType $val(netif) \

```

```

-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-energyModel "EnergyModel"\
-initialEnergy 19906\
-rxPower 0.875\
-txPower 0.18\
-channel $chan_1_
for {set i 0} {$i < $val(nn)} {incr i} {
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ;
}
# Define the nodes positions
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.00
$node_(1) set X_ 5.0
$node_(1) set Y_ 17.0
$node_(1) set Z_ 0.00
$node_(2) set X_ 205.0
$node_(2) set Y_ 5.0
$node_(2) set Z_ 0.00
$node_(3) set X_ 205.0
$node_(3) set Y_ 17.0
$node_(3) set Z_ 0.00
$node_(4) set X_ 405.0
$node_(4) set Y_ 5.0
$node_(4) set Z_ 0.00
$node_(5) set X_ 405.0
$node_(5) set Y_ 17.0
$node_(5) set Z_ 0.00
$node_(6) set X_ 605.0
$node_(6) set Y_ 5.0
$node_(6) set Z_ 0.00
$node_(7) set X_ 605.0
$node_(7) set Y_ 17.0
$node_(7) set Z_ 0.00

```



```
$node_(8) set X_ 805.0
$node_(8) set Y_ 5.0
$node_(8) set Z_ 0.00
$node_(9) set X_ 805.0
$node_(9) set Y_ 17.0
$node_(9) set Z_ 0.00
$node_(10) set X_ 1005.0
$node_(10) set Y_ 5.0
$node_(10) set Z_ 0.00
$node_(11) set X_ 1005.0
$node_(11) set Y_ 17.0
$node_(11) set Z_ 0.00
$node_(12) set X_ 5.0
$node_(12) set Y_ 22.0
$node_(12) set Z_ 0.00
$node_(13) set X_ 5.0
$node_(13) set Y_ 34.0
$node_(13) set Z_ 0.00
$node_(14) set X_ 205.0
$node_(14) set Y_ 22.0
$node_(14) set Z_ 0.00
$node_(15) set X_ 205.0
$node_(15) set Y_ 34.0
$node_(15) set Z_ 0.00
$node_(16) set X_ 405.0
$node_(16) set Y_ 22.0
$node_(16) set Z_ 0.00
$node_(17) set X_ 405.0
$node_(17) set Y_ 34.0
$node_(17) set Z_ 0.00
$node_(18) set X_ 605.0
$node_(18) set Y_ 22.0
$node_(18) set Z_ 0.00
$node_(19) set X_ 605.0
$node_(19) set Y_ 34.0
$node_(19) set Z_ 0.00
$node_(20) set X_ 805.0
$node_(20) set Y_ 22.0
$node_(20) set Z_ 0.00
```

```
$node_(21) set X_ 805.0
$node_(21) set Y_ 34.0
$node_(21) set Z_ 0.00
$node_(22) set X_ 1005.0
$node_(22) set Y_ 22.0
$node_(22) set Z_ 0.00
$node_(23) set X_ 1005.0
$node_(23) set Y_ 34.0
$node_(23) set Z_ 0.00
$node_(24) set X_ 255.0
$node_(24) set Y_ 20.0
$node_(24) set Z_ 0.00
$node_(25) set X_ 755.0
$node_(25) set Y_ 20.0
$node_(25) set Z_ 0.00
#-----
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp0
$udp0 set class_ 0
#Setup a CBR over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 100
$cbr0 set interval_ 0.1
$cbr0 set rate_ 8000
set null0 [new Agent/Null]
$ns_ attach-agent $node_(25) $null0
$ns_ connect $udp0 $null0
#Schedule events for the CBR
$ns_ at 0.0 "$cbr0 start"
$ns_ at 360.0 "$cbr0 stop"
#Setup a UDP connection
```

```
set udp1 [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp1
$udp1 set class_ 0

#Setup a CBR over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 100
$cbr1 set interval_ 0.1
$cbr1 set rate_ 8000

set null1 [new Agent/Null]
$ns_ attach-agent $node_(25) $null1
$ns_ connect $udp1 $null1

#Schedule events for the CBR
$ns_ at 360.0 "$cbr1 start"
$ns_ at 720.0 "$cbr1 stop"

#Setup a UDP connection
set udp2 [new Agent/UDP]
$ns_ attach-agent $node_(2) $udp2
$udp2 set class_ 0

#Setup a CBR over UDP connection
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp2
$cbr2 set packetSize_ 100
$cbr2 set interval_ 0.1
$cbr2 set rate_ 8000

set null2 [new Agent/Null]
$ns_ attach-agent $node_(25) $null2
$ns_ connect $udp2 $null2

#Schedule events for the CBR
$ns_ at 720.0 "$cbr2 start"
$ns_ at 1080.0 "$cbr2 stop"
```



```
#Setup a UDP connection
set udp3 [new Agent/UDP]
$ns_ attach-agent $node_(3) $udp3
$udp3 set class_ 0

#Setup a CBR over UDP connection
set cbr3 [new Application/Traffic/CBR]
$cbr3 attach-agent $udp3
$cbr3 set packetSize_ 100
$cbr3 set interval_ 0.1
$cbr3 set rate_ 8000

set null3 [new Agent/Null]
$ns_ attach-agent $node_(25) $null3
$ns_ connect $udp3 $null3

#Schedule events for the CBR
$ns_ at 1080.0 "$cbr3 start"
$ns_ at 1440.0 "$cbr3 stop"

#Setup a UDP connection
set udp4 [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp4
$udp4 set class_ 0

#Setup a CBR over UDP connection
set cbr4 [new Application/Traffic/CBR]
$cbr4 attach-agent $udp4
$cbr4 set packetSize_ 100
$cbr4 set interval_ 0.1
$cbr4 set rate_ 8000

set null4 [new Agent/Null]
$ns_ attach-agent $node_(25) $null4
$ns_ connect $udp4 $null4

#Schedule events for the CBR
$ns_ at 1440.0 "$cbr4 start"
$ns_ at 1800.0 "$cbr4 stop"
```

```
#Setup a UDP connection
set udp5 [new Agent/UDP]
$ns_ attach-agent $node_(5) $udp5
$udp5 set class_ 0

#Setup a CBR over UDP connection
set cbr5 [new Application/Traffic/CBR]
$cbr5 attach-agent $udp5
$cbr5 set packetSize_ 100
$cbr5 set interval_ 0.1
$cbr5 set rate_ 8000

set null5 [new Agent/Null]
$ns_ attach-agent $node_(25) $null5
$ns_ connect $udp5 $null5

#Schedule events for the CBR
$ns_ at 1800.0 "$cbr5 start"
$ns_ at 2160.0 "$cbr5 stop"

#Setup a UDP connection
set udp6 [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp6
$udp6 set class_ 0

#Setup a CBR over UDP connection
set cbr6 [new Application/Traffic/CBR]
$cbr6 attach-agent $udp6
$cbr6 set packetSize_ 100
$cbr6 set interval_ 0.1
$cbr6 set rate_ 8000

set null6 [new Agent/Null]
$ns_ attach-agent $node_(25) $null6
$ns_ connect $udp6 $null6

#Schedule events for the CBR
$ns_ at 2160.0 "$cbr6 start"
```

```
$ns_ at 2520.0 "$cbr6 stop"  
  
#Setup a UDP connection  
set udp7 [new Agent/UDP]  
$ns_ attach-agent $node_(7) $udp7  
$udp7 set class_ 0
```

```
#Setup a CBR over UDP connection  
set cbr7 [new Application/Traffic/CBR]  
$cbr7 attach-agent $udp7  
$cbr7 set packetSize_ 100  
$cbr7 set interval_ 0.1  
$cbr7 set rate_ 8000
```

```
set null7 [new Agent/Null]  
$ns_ attach-agent $node_(25) $null7  
$ns_ connect $udp7 $null7
```

```
#Schedule events for the CBR  
$ns_ at 2520.0 "$cbr7 start"  
$ns_ at 2880.0 "$cbr7 stop"
```

```
set udp8 [new Agent/UDP]  
$ns_ attach-agent $node_(8) $udp8  
$udp8 set class_ 0
```

```
#Setup a CBR over UDP connection  
set cbr8 [new Application/Traffic/CBR]  
$cbr8 attach-agent $udp8  
$cbr8 set packetSize_ 100  
$cbr8 set interval_ 0.1  
$cbr8 set rate_ 8000
```

```
set null8 [new Agent/Null]  
$ns_ attach-agent $node_(25) $null8  
$ns_ connect $udp8 $null8
```

```
#Schedule events for the CBR  
$ns_ at 2880.0 "$cbr8 start"
```

```

$ns_ at 3240.0 "$cbr8 stop"
set udp9 [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp9
$udp9 set class_ 0

#Setup a CBR over UDP connection
set cbr9 [new Application/Traffic/CBR]
$cbr9 attach-agent $udp9
$cbr9 set packetSize_ 100
$cbr9 set interval_ 0.1
$cbr9 set rate_ 8000

set null9 [new Agent/Null]
$ns_ attach-agent $node_(25) $null9
$ns_ connect $udp9 $null9

#Schedule events for the CBR
$ns_ at 3240.0 "$cbr9 start"
$ns_ at 3600.0 "$cbr9 stop"

#Setup a UDP connection
set udp10 [new Agent/UDP]
$ns_ attach-agent $node_(10) $udp10
$udp10 set class_ 0

#Setup a CBR over UDP connection
set cbr10 [new Application/Traffic/CBR]
$cbr10 attach-agent $udp10
$cbr10 set packetSize_ 100
$cbr10 set interval_ 0.1
$cbr10 set rate_ 8000

set null10 [new Agent/Null]
$ns_ attach-agent $node_(25) $null10
$ns_ connect $udp10 $null10

#Schedule events for the CBR
$ns_ at 0.0 "$cbr10 start"

```

```

$ns_ at 360.0 "$cbr10 stop"

#Setup a UDP connection
set udp11 [new Agent/UDP]
$ns_ attach-agent $node_(11) $udp11
$udp11 set class_ 0

#Setup a CBR over UDP connection
set cbr11 [new Application/Traffic/CBR]
$cbr11 attach-agent $udp11
$cbr11 set packetSize_ 100
$cbr11 set interval_ 0.1
$cbr11 set rate_ 8000

set null11 [new Agent/Null]
$ns_ attach-agent $node_(25) $null11
$ns_ connect $udp11 $null11

#Schedule events for the CBR
$ns_ at 0.0 "$cbr11 start"
$ns_ at 360.0 "$cbr11 stop"

#-----
#Setup a UDP connection
set udp12 [new Agent/UDP]
$ns_ attach-agent $node_(22) $udp12
$udp12 set class_ 0

#Setup a CBR over UDP connection
set cbr12 [new Application/Traffic/CBR]
$cbr12 attach-agent $udp12
$cbr12 set packetSize_ 100
$cbr12 set interval_ 0.1
$cbr12 set rate_ 8000

set null12 [new Agent/Null]
$ns_ attach-agent $node_(24) $null12
$ns_ connect $udp12 $null12

```

```

#Schedule events for the CBR
$ns_ at 0.0 "$cbr12 start"
$ns_ at 360.0 "$cbr12 stop"

#Setup a UDP connection
set udp13 [new Agent/UDP]
$ns_ attach-agent $node_(23) $udp13
$udp13 set class_ 0

#Setup a CBR over UDP connection
set cbr13 [new Application/Traffic/CBR]
$cbr13 attach-agent $udp13
$cbr13 set packetSize_ 100
$cbr13 set interval_ 0.1
$cbr13 set rate_ 8000

set null13 [new Agent/Null]
$ns_ attach-agent $node_(24) $null13
$ns_ connect $udp13 $null13

#Schedule events for the CBR
$ns_ at 0.0 "$cbr13 start"
$ns_ at 360.0 "$cbr13 stop"

#Setup a UDP connection
set udp14 [new Agent/UDP]
$ns_ attach-agent $node_(20) $udp14
$udp14 set class_ 0

#Setup a CBR over UDP connection
set cbr14 [new Application/Traffic/CBR]
$cbr14 attach-agent $udp14
$cbr14 set packetSize_ 100
$cbr14 set interval_ 0.1
$cbr14 set rate_ 8000

set null14 [new Agent/Null]
$ns_ attach-agent $node_(24) $null14
$ns_ connect $udp14 $null14

```

```

#Schedule events for the CBR
$ns_ at 0.0 "$cbr14 start"
$ns_ at 360.0 "$cbr14 stop"

#Setup a UDP connection
set udp15 [new Agent/UDP]
$ns_ attach-agent $node_(21) $udp15
$udp15 set class_ 0

#Setup a CBR over UDP connection
set cbr15 [new Application/Traffic/CBR]
$cbr15 attach-agent $udp15
$cbr15 set packetSize_ 100
$cbr15 set interval_ 0.1
$cbr15 set rate_ 8000

set null15 [new Agent/Null]
$ns_ attach-agent $node_(24) $null15
$ns_ connect $udp15 $null15

#Schedule events for the CBR
$ns_ at 0.0 "$cbr15 start"
$ns_ at 360.0 "$cbr15 stop"

#Setup a UDP connection
set udp16 [new Agent/UDP]
$ns_ attach-agent $node_(18) $udp16
$udp16 set class_ 0

#Setup a CBR over UDP connection
set cbr16 [new Application/Traffic/CBR]
$cbr16 attach-agent $udp16
$cbr16 set packetSize_ 100
$cbr16 set interval_ 0.1
$cbr16 set rate_ 8000

set null16 [new Agent/Null]
$ns_ attach-agent $node_(24) $null16

```

```
$ns_ connect $udp16 $null16
#Schedule events for the CBR
$ns_ at 0.0 "$cbr16 start"
$ns_ at 360.0 "$cbr16 stop"

#Setup a UDP connection
set udp17 [new Agent/UDP]
$ns_ attach-agent $node_(19) $udp17
$udp17 set class_ 0

#Setup a CBR over UDP connection
set cbr17 [new Application/Traffic/CBR]
$cbr17 attach-agent $udp17
$cbr17 set packetSize_ 100
$cbr17 set interval_ 0.1
$cbr17 set rate_ 8000

set null17 [new Agent/Null]
$ns_ attach-agent $node_(24) $null17
$ns_ connect $udp17 $null17

#Schedule events for the CBR
$ns_ at 0.0 "$cbr17 start"
$ns_ at 360.0 "$cbr17 stop"

#Setup a UDP connection
set udp18 [new Agent/UDP]
$ns_ attach-agent $node_(16) $udp18
$udp18 set class_ 0

#Setup a CBR over UDP connection
set cbr18 [new Application/Traffic/CBR]
$cbr18 attach-agent $udp18
$cbr18 set packetSize_ 100
$cbr18 set interval_ 0.1
$cbr18 set rate_ 8000

set null18 [new Agent/Null]
```



```
$ns_ attach-agent $node_(24) $null18
$ns_ connect $udp18 $null18
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr18 start"
$ns_ at 360.0 "$cbr18 stop"
```

```
#Setup a UDP connection
set udp19 [new Agent/UDP]
$ns_ attach-agent $node_(17) $udp19
$udp19 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr19 [new Application/Traffic/CBR]
$cbr19 attach-agent $udp19
$cbr19 set packetSize_ 100
$cbr19 set interval_ 0.1
$cbr19 set rate_ 8000
```

```
set null19 [new Agent/Null]
$ns_ attach-agent $node_(24) $null19
$ns_ connect $udp19 $null19
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr19 start"
$ns_ at 360.0 "$cbr19 stop"
```

```
#Setup a UDP connection
set udp20 [new Agent/UDP]
$ns_ attach-agent $node_(14) $udp20
$udp20 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr20 [new Application/Traffic/CBR]
$cbr20 attach-agent $udp20
$cbr20 set packetSize_ 100
$cbr20 set interval_ 0.1
$cbr20 set rate_ 8000
```

```
set null20 [new Agent/Null]
$ns_ attach-agent $node_(24) $null20
$ns_ connect $udp20 $null20
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr20 start"
$ns_ at 360.0 "$cbr20 stop"
```

```
#Setup a UDP connection
set udp21 [new Agent/UDP]
$ns_ attach-agent $node_(15) $udp21
$udp21 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr21 [new Application/Traffic/CBR]
$cbr21 attach-agent $udp21
$cbr21 set packetSize_ 100
$cbr21 set interval_ 0.1
$cbr21 set rate_ 8000
```

```
set null21 [new Agent/Null]
$ns_ attach-agent $node_(24) $null21
$ns_ connect $udp21 $null21
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr21 start"
$ns_ at 360.0 "$cbr21 stop"
```

```
#Setup a UDP connection
set udp22 [new Agent/UDP]
$ns_ attach-agent $node_(12) $udp22
$udp22 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr22 [new Application/Traffic/CBR]
$cbr22 attach-agent $udp22
$cbr22 set packetSize_ 100
$cbr22 set interval_ 0.1
$cbr22 set rate_ 8000
```

```

set null122 [new Agent/Null]
$ns_ attach-agent $node_(24) $null122
$ns_ connect $udp22 $null122

#Schedule events for the CBR
$ns_ at 0.0 "$cbr22 start"
$ns_ at 360.0 "$cbr22 stop"

#Setup a UDP connection
set udp23 [new Agent/UDP]
$ns_ attach-agent $node_(13) $udp23
$udp23 set class_ 0

#Setup a CBR over UDP connection
set cbr23 [new Application/Traffic/CBR]
$cbr23 attach-agent $udp23
$cbr23 set packetSize_ 100
$cbr23 set interval_ 0.1
$cbr23 set rate_ 8000

set null123 [new Agent/Null]
$ns_ attach-agent $node_(24) $null123
$ns_ connect $udp23 $null123

#Schedule events for the CBR
$ns_ at 0.0 "$cbr23 start"
$ns_ at 360.0 "$cbr23 stop"
#-----

# defines the node size in nam
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 100
}
# Tell nodes simulation ends at 60.0
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 60.0 "$node_($i) reset";
}
$ns_ at 10.0 "stop"

```

```
$ns_ at 10.0 "puts \"\nNS EXITING...\n\""  
$ns_ at 10.01 "$ns_ halt"  
proc stop {} {  
    global ns_ tesis200 tesis200  
    $ns_ flush-trace  
    close $tesis200  
    #set hasDISPLAY 0  
    exec nam tesis200.nam &  
}  
puts "\nStarting Simulation..."  
$ns_ run
```

LAMPIRAN B



Introduction

The MMA7361 from Freescale is a very nice sensor with easy analog interface. The MMA7361 is a 3.3V part and outputs an analog voltage for each of the three outputs. This voltage is in ratio to the measured acceleration and to the supply voltage (ratiometric). It has selectable sensitivity by dip switch. You will need some extra hardware to convert this analog signal to a usable digital one. The Arduino is really good option for it. This break board is especially designed for Arduino which has 3 JST connector that can be easily plug into our IO/Sensor expansion board.

Specification

- Voltage:3.3-8V
- Selectable sensitivity:±1. 5g/6g
- Low power:500 μ A @ measurement mode, 3 μ A @standby ;
- High sensivity: 800 mV/g @ 1.5g ;
- Interface:Analog Output
- Low pass filter
- Size:23x26mm
- Weight: 5 gram

LAMPIRAN C

Lampiran Program Koordinator :

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  while (Serial.available() )
  {
    Serial.write(Serial.read()); // reply with whatever you receive
  }
}
```

Lampiran Program sensor:

```
#define TIME_HEADER "T"
#define TIME_REQUEST 7

int x2pin = 7;
int y2pin = 8;
int z2pin = 9;
float x2,y2,z2;

float vx, vy, vz, gx, gy, gz;
byte pesan[1];

byte b[30];
byte ACK[2];

int i;
int x,y;
```

```

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);}
void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("@");
  Serial.print(" ");
  Serial.print("SENSOR1");
  Serial.print(" ");
  //digitalClockDisplay();
  //Serial.print(" ");
  dataSensor();
  Serial.println("#");
  delay(250); //<<<<<<<<<<<<<<< DELAY SAMPLING
}
void digitalClockDisplay () {
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
}
void printDigits(int digits){
  Serial.print(":");
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);

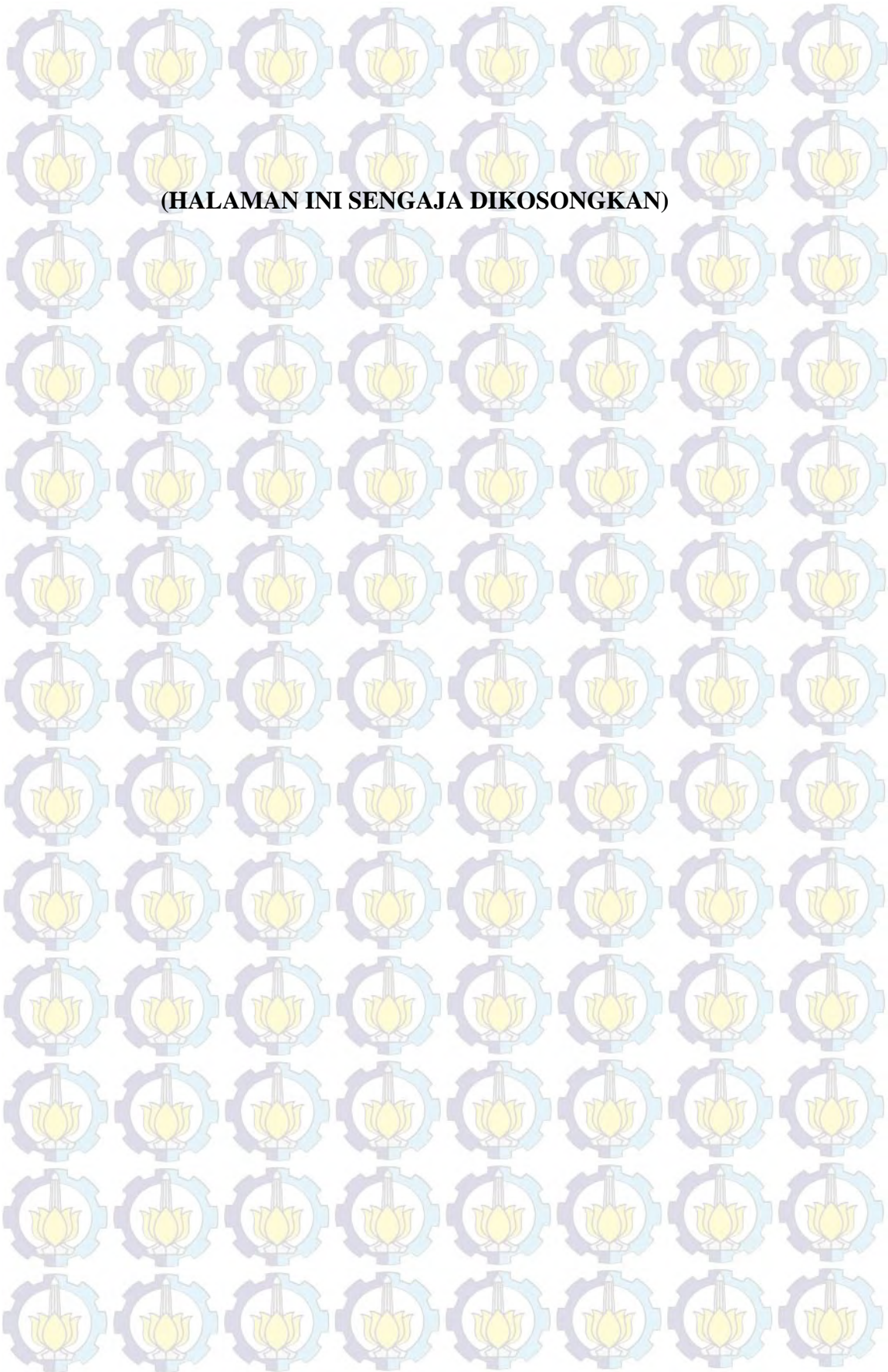
```

```
}  
void dataSensor(){  
  x2=analogRead(x2pin);  
  y2=analogRead(y2pin);  
  z2=analogRead(z2pin);  
  analogReadResolution(16);  
  //x2=((z/256)*5); //untuk 6g 0.206 1.5 g 0.8  
  vx = (x2*5/65520)-2.50;  
  gx = vx/0.206;  
  vy = (y2*5/65520)-2.8;  
  gy = vy/0.206;  
  vz = (z2*5/65520)-3.3;  
  gz = vz/0.206;  
  digitalClockDisplay(); // jam  
  Serial.print(" ");  
  Serial.print(gx,4);  
  Serial.print(" ");  
  Serial.print(gy,4);  
  Serial.print(" ");  
  Serial.print(gz,4);  
  Serial.print(" ");  
}
```


Spesifikasi Modul RF XBee-PRO

Tabel 1 Spesifikasi Modul RF XBee-PRO

Performance	
Indoor Urban-Range	up to 300' (100 m)
Outdoor RF line-of-sight Range	up to 1 mile (1500 m)
Transmit Power Output (software selectable)	60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 – 115200 bps (non-standard baud rates also supported)
Receiver Sensitivity	- 100 dBm (1% packet error rate)
Power Requirements	
Supply Voltage	2.8 – 3.4 V
Idle / Receive Current (typical)	55 mA (@3.3 V)
Power-down Current	< 10 μ A
General	
Operating Frequency	ISM 2.4 GHz
Frequency Band	2.4 - 2.4835 GHz
Modulation	OQPSK
Dimensions	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector
Networking & Security	
Supported Network Topologies	Point-to-point, Point-to multipoint & Peer-to-peer
Number of Channels	12 Direct Sequence Channels



BAB 1

PENDAHULUAN

1.1. Latar Belakang

Beberapa bangunan memiliki kerentanan terhadap kerusakan, sebagai contoh jembatan, bendungan, bandara dan stadion. Kerusakan ini tidak mudah untuk diantisipasi sehingga terkadang dapat menimbulkan korban. Pada tahun 2011, telah terjadi peristiwa robohnya sebuah jembatan di Indonesia, yakni Jembatan Kutai Kartanegara yang memiliki panjang 720 meter dan mengakibatkan banyak korban [1]. Peristiwa runtuhnya jembatan ini disebabkan oleh adanya tegangan yang berlebihan terhadap jembatan tersebut.

Kerusakan suatu jembatan umumnya disebabkan oleh beberapa faktor. Faktor ini dapat dikerucutkan menjadi 2 tipe yakni faktor internal dan faktor eksternal [2]. Faktor internal meliputi kerusakan yang disebabkan oleh komponen penyusun dari jembatan tersebut. Sedangkan faktor eksternal meliputi kerusakan yang disebabkan oleh alam sekitar.

Untuk meminimalisir terjadinya kerusakan jembatan maka perlu diterapkan suatu sistem dan teknologi pemantauan. Teknologi ini adalah Bridges Structural Health Monitoring System atau bisa disebut Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan. Teknologi SMKS ini memiliki potensi yang sangat baik untuk meningkatkan operasi rutin dan pemeliharaan struktur pada jembatan [2]. Teknologi ini dapat memperpanjang usia dari jembatan karena dapat mendeteksi lebih awal. Teknologi ini menggunakan sensor-sensor. Adapun sensor-sensor yang digunakan diantaranya sensor gerakan dan getaran (accelerometer), sensor tekanan (pressure gauge), sensor suhu (thermocouple), dan sensor akustik (piezoelectric) [3]. Sensor-sensor tersebut bertugas untuk mengambil data sesuai dengan jenis sensor, kemudian data dari sensor tersebut akan dikirim menuju pusat, sehingga pusat akan mengetahui kondisi terkini dari jembatan tersebut.

Pada SMKS terdapat beberapa tingkatan dalam penilaian kondisi jembatan yaitu identifikasi anomali dan kerusakan dalam struktur, lokalisasi kerusakan, kuantifikasi keparahan kerusakan, dan prediksi sisa umur layanan struktur [2]. Tingkatan ini bertujuan untuk melakukan pencegahan dan mendeteksi dengan cepat masalah yang ditemukan.

Untuk proses komunikasi yang digunakan pada SMKS di bagi menjadi 2 jenis yakni komunikasi berbasis kabel dan nirkabel. Komunikasi berbasis kabel memiliki kelebihan yaitu data yang dikirimkan dari sensor sangat akurat dan juga tidak memerlukan daya yang cukup banyak karena dayanya diambil langsung dari pusat. Akan tetapi komunikasi berbasis kabel ini terkendala pada instalasi kabel sehingga semakin jauh jarak jangkauannya maka memerlukan kabel yang cukup banyak yang juga berpengaruh pada biaya instalasinya. Menurut [4] sekitar 25 persen dari total biaya anggaran SMKS digunakan untuk instalasi kabel. Meninjau hal ini, maka perlu adanya sistem monitoring yang tidak memerlukan instalasi kabel yakni menggunakan jaringan sensor nirkabel [5]. Komunikasi nirkabel tersebut atau lebih dikenal dengan wireless sensor network (WSN). WSN mempunyai kelebihan yaitu tidak memerlukan komponen tambahan seperti saluran kabel, sensor mudah diganti jika mengalami kerusakan, mudah dikonfigurasi ulang, dan dengan sistem Ad-Hoc dan Multi-Hop komunikasi data menjadi lebih praktis.

Dalam pemantauan kesehatan jembatan yang menggunakan WSN pasti memerlukan konsumsi daya agar sensor-sensor dapat bekerja dengan baik. Namun juga perlu di pertimbangkan akan daya yang digunakan, dimana perlu adanya teknik atau metode yang digunakan agar konsumsi daya yang digunakan lebih efisien. Sedangkan WSN memiliki keterbatasan daya dimana rata-rata sensor dicatu oleh baterai yang mempunyai *lifetime* sangat terbatas. Untuk peralatan komunikasi yang digunakan untuk mengirimkan data melalui jaringan nirkabel harus memperhatikan ketersediaan bandwidth, range sensor dan range komunikasi sensor [6]. Pada penelitian [7] terdapat beberapa teknik untuk menghemat daya secara umum, yakni:

1. Penjadwalan node

2. Mengatur control daya dengan menyesuaikan jangkauan transmisi di sekitar node

3. Mengurangi jumlah data yang ditransmisikan dan menghindari aktivitas yang sia-sia.

Selain itu langkah penghematan daya pada WSN bertujuan agar masa hidup sensor lebih lama sehingga penggunaan sensor dapat lebih maksimal dan dapat digunakan untuk proses monitoring secara kontinyu. Pada [8] mengusulkan penghematan daya dengan mengatur topologi yang digunakan dengan menfokuskan pada topologi cluster-tree dengan berbasis simulasi. Daya yang digunakan lebih efisien karena topologi disesuaikan dengan lingkungan yang akan diamati, namun studi ini tidak mempertimbangkan konteks implementasi jaringan secara *real time*.

Pada penelitian ini, teknik manajemen konsumsi daya didapatkan melalui pengaturan node-node sensor dengan memanfaatkan kondisi *sleep/awake* pada node sensor tersebut. Node *awake* ketika terjadi proses pentransmisian data dengan node tetangganya. Namun ketika node tetangga mengalami penurunan daya, maka node yang sedang mentransmisikan data akan mencari node tetangga lain yang masih memiliki daya yang cukup banyak. Sedangkan untuk node yang tidak mentransmisikan data dalam kondisi *sleep*. Pemilihan node dalam proses pentransmisian data mempertimbangkan energi, jarak, *packet loss, throughput* serta *delay* pada node yang akan dituju.

Dalam pemilihan node untuk proses pentransmisian data dipengaruhi oleh rute atau jalur yang akan dilalui. Rute yang tepat akan menjadikan proses transmisi lebih maksimal. Node-node yang dipilih dalam proses transmisi berperan tidak hanya menjadi pengirim dan penerima, namun juga berperan sebagai penunjang node yang lainnya. Maka dari itu diperlukan routing protokol untuk menunjang proses pentransmisian antar node-node. Pada jaringan sensor nirkabel memanfaatkan sistem Ad-Hoc dan MultiHop.

Pada sistem Ad-hoc terdapat jenis routing protokol diantaranya AODV dan DSDV. Beberapa penelitian telah melakukan pengamatan tentang routing protokol untuk jaringan ad-hoc. Pada studi [9] melakukan penelitian tentang routing protokol AODV dengan memanfaatkan trafik CBR dan TCP, dimana

untuk pengiriman paket menggunakan Packet Delivery Fraction (PDF). Pada TCP terdapat proses pengiriman ulang untuk paket yang mengalami error sedangkan CBR tidak ada pengiriman ulang sehingga banyaknya paket yang diterima lebih kecil karena adanya kemungkinan error. Menurut studi [10] kinerja routing protokol DSDV (Destination Sequenced Distance Vector) pada jaringan wireless dipengaruhi oleh lama waktu dan kecepatan, dimana ketika semakin lama waktu dan semakin sedikit kecepatan maka kinerja DSDV semakin tinggi. Penelitian ini berbasis simulasi menggunakan Network Simulator-2 (NS2). Network simulator ini merupakan salah satu simulator berbasis open source yang digunakan untuk membantu analisa dalam hal pemodelan media, protocol dan trafiknya.

Penelitian ini berfokus pada desain dan analisa untuk manajemen konsumsi daya dengan cara mengkondisikan sleep/awake sensor pada node-node melalui pengaturan topologi serta routing protokolnya. Sehingga hasil simulasi dapat diimplementasikan pada Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.

1.2. Rumusan Masalah

Pada penelitian ini, rumusan permasalahan yang diambil adalah :

1. Bagaimana manajemen daya pada node untuk mendapatkan konsumsi daya yang lebih efisien.
2. Routing manakah yang baik untuk diterapkan pada Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.
3. Topologi apa yang sesuai untuk di terapkan pada Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.

1.3. Batasan Masalah

Pada penelitian ini, batasan masalah yang diambil adalah :

1. Simulasi menggunakan NS-2.
2. Topologi yang digunakan adalah star, tree dan mesh
3. Wireless module yang digunakan adalah Protocol 802.15.4 dan mengacu pada datasheet XBee Pro versi v1.
4. Routing protokol menggunakan DSDV dan AODV.

5. Sensor yang digunakan adalah sensor sensor accelerometer
6. Protocol 802.15.4 memiliki coverage area 250 m.

1.4. Tujuan

Tujuan dari penelitian ini adalah :

- 1) Memperoleh nilai throughput, delay, paket dan energi untuk setiap node pada saat node berkomunikasi.
- 2) Mendapatkan topologi dan routing terbaik untuk diterapkan pada Sistem Monitoring Struktur (SMKS) Jembatan.

1.5. Manfaat

Dari usulan penelitian tesis ini, diharapkan dapat memberikan kontribusi dalam hal manajemen konsumsi daya pada WSN untuk Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan.

1.6. Sistematika Penulisan

Untuk memudahkan pembahasan, usulan tesis ini disusun dengan sistematika sebagai berikut :

BAB 1. PENDAHULUAN

Bagian pendahuluan berisi latar belakang, rumusan masalah, batasan masalah, metode penelitian yang digunakan, dan sistematika penulisan. Bagian ini merupakan hal-hal yang mendasari dan pentingnya penelitian yang dilakukan.

BAB 2. TINJAUAN PUSTAKA

Pada bab ini berisi tinjauan pustaka yang mendukung dan berkaitan dengan penelitian yaitu tentang Sistem Monitoring Kesehatan Struktur (SMKS), Wireless Sensor Sensor Network (WSN), XBee Pro versi v1, Arduino Due, Ad hoc On demand Distance Vector (AODV), Destination Sequenced Distance Vector (DSDV) dan Network Simulator-2 (NS-2).

BAB 3. METODE PENELITIAN

Pada bab ini berisi metode penelitian yang digunakan dalam hal pengimplementasian manajemen konsumsi daya yang diperlukan. Komunikasi sensor node disimulaikan dengan NS-2 berdasarkan asumsi-asumsi yang dipaparkan pada bab ini serta menggunakan parameter dari datasheet XBee Pro versi 1.

BAB 4. HASIL DAN ANALISA

Pada bab ini berisi tentang hasil simulasi yang telah dilakukan beserta analisisnya. Simulasi yang dilakukan adalah berdasarkan desain yang telah dikemukakan pada bab 3 dengan menggunakan simulator NS-2 dan menggunakan parameter dari datasheet XBee Pro versi v1.

BAB 5. KESIMPULAN DAN SARAN

Pada bab ini berisi tentang kesimpulan penelitian dari analisa bab 3 beserta saran untuk kemajuan penelitian ini ke depannya

BAB 2

TINJAUAN PUSTAKA

2.1. Sistem Monitoring Kesehatan Struktur (SMKS)

Suatu bangunan memiliki masa ketahanan dalam beroperasi. Namun tidak menutup kemungkinan adanya penurunan atas kemampuan ketahanan yang dimiliki oleh bangunan tersebut, meskipun bangunan tersebut didesain agar dapat beroperasi untuk jangka waktu yang cukup lama. Tidak menutup kemungkinan penurunan performa mengakibatkan kerusakan pada badan bangunan, sehingga memerlukan biaya perbaikan yang tidak sedikit.

Berkurangnya kemampuan fisik dinilai dari perhitungan terhadap kondisi fisik kesehatan bangunan tersebut. Penilaian ini perlu dilakukan secara terus menerus agar dapat dilakukan suatu tindakan untuk mencegah terjadinya suatu hal yang tidak diinginkan. Maka dari itu perlu adanya suatu sistem untuk pemantauan kesehatan bangunan tersebut.

Seiring semakin majunya teknologi dalam bidang instrumentasi dan juga dalam bidang teknologi informasi dan komunikasi, maka sistem monitoring ini dapat dijalankan dengan lebih mudah. Terdapat suatu bidang baru dalam hal pemantauan kesehatan struktur dari kerusakan yakni Sistem Monitoring Kesehatan Struktur (SMKS).

2.1.1. SMKS Jembatan

Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan merupakan bidang pemantauan pada jembatan, dimana teknologi ini dapat mempepanjang umur pengoperasian jembatan karena dapat mendeteksi lebih awal tentang adanya kerusakan sehingga meminimalisir terjadinya kerusakan yang lebih parah. Beberapa tujuan dari SMKS ini adalah sebagai berikut [13]:

1. Menjamin keamanan struktur
2. Memperoleh perencanaan pemeliharaan struktur yang rasional dan ekonomis
3. Mencapai pekerjaan pemeliharaan yang aman dan ekonomis
4. Mengidentifikasi penyebab respon yang tidak dapat di terima

SMKS digunakan untuk pemantauan kondisi fisik jangka pendek maupun jangka panjang. Dalam praktiknya SMKS ini menggunakan beberapa sensor yang diletakkan pada badan jembatan yang kemudian digunakan untuk memperoleh respon dari struktur jembatan tersebut. Selanjutnya respon dari sensor akan di analisis untuk memperoleh informasi tentang tingkat kerusakannya. Terdapat beberapa faktor yang berkaitan dalam hal pengembangan dan kemajuan teknologi SMKS ini, yakni [14]:

1. Kemajuan terbaru dalam teknologi penginderaan dengan kecepatan tinggi dan sirkuit elektronik murah, dan pengembangan validasi sinyal yang sangat efisien dan pengolahan metode.
2. Perkembangan yang sedang berlangsung dalam teknologi komunikasi, yaitu banyak penggunaan internet dan teknologi nirkabel.
3. Perkembangan transmisi data dan sistem pengumpulan, serta sistem pengarsipan dan pengambilan data.
4. Kemajuan dalam pengolahan data, termasuk model deteksi kerusakan dan algoritma kecerdasan buatan.

2.1.2. Komponen Sistem Monitoring

Pada sistem monitoring mengacu pada pemantauan terus menerus menggunakan sensor yang ada, baik sensor yang tertanam pada badan jembatan ataupun yang melekat pada bagian luarnya. Sistem monitoring terdiri dari enam komponen, diantaranya :

1. Data Akuisisi

Meliputi berbagai jenis sensor dan pengumpulan data yang diinginkan.

2. Komunikasi data

Meliputi transmisi data dari site ke storage dan analisis lokasi. Sebagai contoh pengolahan informasi data jarak jauh.

3. Pengolahan dan analisis data

Meliputi pembersihan data dari gangguan dan informasi tambahan.

4. Penyimpanan data yang diolah

Data yang di peroleh disimpan untuk keperluan tahap selanjutnya.

5. Analisa diagnosa

Meliputi analisa pendeteksian kerusakan dan konversi data baru ke dalam respon structural.

6. Pengambilan informasi sesuai yang diperlukan

Meliputi pengambilan informasi dari hasil analisa diagnosa yang telah dilakukan.

2.1.3. Syarat Monitoring Jembatan

Pada sistem monitoring untuk mendapatkan data respon struktur sepanjang periode pengukuran untuk memverifikasi parameter beban stokastik dan respon struktur yang akan dibandingkan dengan respon yang dihitung.

Beberapa data digunakan untuk melihat kebenaran dari struktur atau untuk memverifikasi kekurangan yang ada. Monitoring dalam jangka waktu yang pendek meliputi pembebanan pada struktur atau monitoring beban yang tidak diharapkan. Contoh-contoh monitoring, diantaranya [13]:

1. Respon Stokastik

Karakteristik gempa, angin dan beban lalu lintas dan respon struktur dapat diukur di lapangan untuk memverifikasi prediksi yang dibuat pada model numerik pada tahap desain.

2. Beban Internal

Selain pengukuran permanen, pengukuran intensif dapat diulangi berkali-kali menggunakan mobile sensor untuk memetakan perubahan dalam distribusi gaya pada cable-stayed, tiang fondasi, dll. Distribusi regangan dapat dimonitor pada periode yang panjang untuk mengukur perubahan distribusi tegangan.

3. Respon *Fatigue* /Kelelahan

Beban fatigue pada sambungan las, dek dan balok diukur dengan strain gauge atau sistem accelerometer.

4. Respon Deterministik

Perpindahan pada buffer hidraulik, dampers, siar muai (expansion joint) yang tergantung dari temperatur dan distribusi beban pada dek ortotropik dapat dimonitor oleh sensor temperatur dan sensor displacement, tiltmeters dan sistem GPS.

5. *Global Static Response*

Respon statik pada fondasi, rangkai (creep) dan penyusutan (shrinkage), distribusi regangan pada kabel utama dapat dimonitor oleh sensor khusus.

Pengukuran dapat dilakukan untuk mengkalkulasikan seperti temperatur/regangan rata-rata dan perbedaan temperatur/regangan pada jarak yang jauh.

2.1.4. Klasifikasi Monitoring

SMKS dapat diklasifikasikan menjadi beberapa macam, diantaranya adalah klasifikasi menurut tingkat kecanggihannya, jenis informasinya dan pengambilan keputusan algoritma yang diberikan. Menurut [14] sistem monitoring berdasarkan tingkat kecanggihan dan kemajuan dapat di klasifikasi sebagai berikut:

1. Tingkat I

Pada tingkat 1 ini sistem monitoring dapat mendeteksi kerusakan dalam struktur, akan tetapi tidak dapat memberikan informasi tentang sifat, lokasi ataupun keparahan kerusakan. Sehingga pada tingkat ini belum bisa diambil penilaian tentang kesehatan dari suatu struktur tersebut.

2. Tingkat II

Untuk tingkat II ini sedikit lebih baik dari sistem pada tingkat I, pada tingkat ini dapat mendeteksi adanya kerusakan dan juga memberikan informasi tentang lokasi kerusakan.

3. Tingkat III

Pada tingkat III ini dapat mendeteksi dan menentukan kerusakan dan dapat memberikan beberapa indikasi dari beratnya.

4. Tingkat IV

Tingkat IV merupakan tingkatan yang paling canggih karena mampu memberikan informasi rinci tentang keberadaan, lokasi dan tingkat keparahan kerusakan sehingga informasi yang di dapat digunakan untuk mengevaluasi keamanan dari sistem.



Gambar 2.1 Klasifikasi sistem Monitoring Berdasarkan Kemampuan Pendeteksian dan Pengambilan Keputusan [14]

2.1.5. Parameter Akuisisi Data Dalam Monitoring Jembatan

Dalam pemantauan terdapat beberapa parameter penting yang perlu dipertimbangkan. Berikut ini menjelaskan beberapa parameter penting yang biasanya diperlukan untuk kebanyakan pengujian dan monitoring jembatan [13]:

1. Jumlah Sensor

Dalam setiap aplikasi pengujian atau pemantauan, jumlah sensor akan tergantung pada ukuran dan kompleksitas dari struktur dan pada kompleksitas perilaku yang sedang dievaluasi. Hal ini tidak biasa untuk aplikasi pengujian dan monitoring skala besar dan bahkan beberapa jembatan ukuran sedang yang memanfaatkan 120 atau lebih sensor. Karakteristik kinerja perangkat keras akuisisi data sering kali tergantung pada jumlah saluran sensor sedang dipindai.

2. Tipe Sensor

Jenis-jenis sensor yang akan digunakan akan menentukan banyak karakteristik yang dibutuhkan dari perangkat keras akuisisi data. Sebagai contoh, sebuah pengukur regangan resistansi memiliki karakteristik persyaratan output spesifik dan pengkondisian sinyal. Akuisisi data biasanya akan bervariasi untuk setiap jenis sensor yang digunakan.

Selanjutnya, dalam sebagian besar aplikasi, lebih dari satu jenis sensor akan dibutuhkan. Hal ini penting untuk memastikan bahwa karakteristik perangkat keras akuisisi data yang sesuai persyaratan untuk setiap jenis sensor sehingga pengukuran yang paling akurat dan diandalkan dapat diperoleh. Hal ini sering membutuhkan perangkat keras akuisisi data yang

cukup fleksibel untuk mengakomodasi berbagai bentuk peng-kondisian sinyal. Dalam beberapa kasus, penggunaan lebih dari satu jenis sistem akuisisi data dapat menawarkan solusi yang paling fleksibel dan pembiayaan yang efektif.

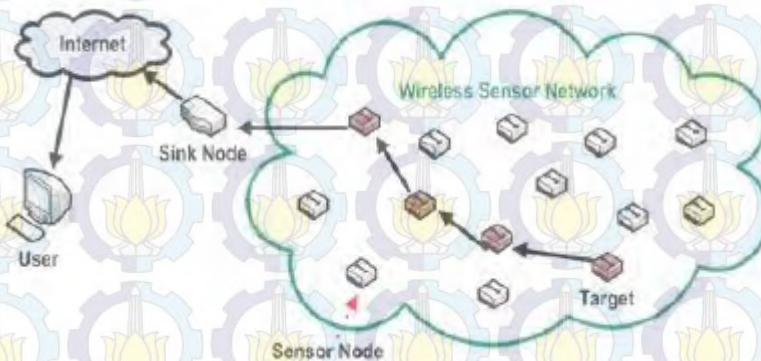
3. Mode Data Akuisisi

Ada dua tipe dasar mode akuisisi data, polled dan event-based. Pada mode polled, sistem akuisisi data umumnya dioperasikan secara terus menerus dan sinyal sensor yang sesuai dikumpulkan dan dicatat secara waktu berkala. Sensor yang mengukur variabel fisik yang statis atau yang perlahan-lahan bervariasi dengan waktu yang sesuai untuk mode polled akuisisi data. Sedangkan pada akuisisi data event-based, sinyal sensor dikumpulkan hanya selama terjadinya beberapa peristiwa penting yang ditetapkan pengguna. Peristiwa penting ini dapat acak atau deterministik dan beberapa contoh umum dari peristiwa tersebut meliputi kegiatan konstruksi, kontrol tes, sebuah truk berat melintasi jembatan atau suatu periode dengan volume lalu lintas berat, kecelakaan, dan kondisi lingkungan normal atau bencana alam seperti banjir, gempa bumi atau badai. Variabel fisik yang diukur oleh sensor selama terjadinya peristiwa seperti itu dapat berubah lambat atau cepat seiring waktu, tergantung pada kecepatan peristiwa.

2.2. *Wireless Sensor Network (WSN)*

Wireless Sensor Network (WSN) atau Jaringan Sensor Nirkabel merupakan suatu jaringan yang terdiri dari beberapa sensor yang masing-masing sensor tersebut memiliki kemampuan untuk merasakan (sensing), memproses serta berkomunikasi. Pada WSN, node sensor disebar dengan tujuan untuk menangkap adanya gejala atau fenomena yang hendak diteliti. Jumlah node yang disebar dapat ditentukan sesuai kebutuhan dan tergantung beberapa faktor misalnya luas area, kemampuan sensing} node, dan sebagainya. Tiap node memiliki kemampuan untuk mengumpulkan data dan meroutingkannya kembali ke Base Station}. Node sensor dapat mengumpulkan data dalam jumlah yang

besar dari gejala yang timbul dari lingkungan sekitar. untuk arsitektur WSN secara umum dapat direpresentasikan sebagai berikut:



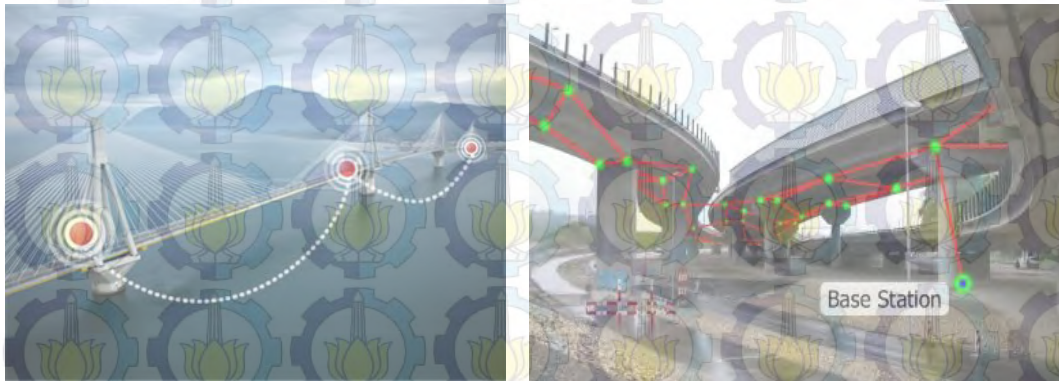
Gambar 2.2. Arsitektur WSN secara umum [15]

Pada Gambar 2.2, dapat dilihat, node sensor yang berukuran kecil disebar dalam di suatu area sensor. Node sensor tersebut memiliki kemampuan untuk merutekan data yang dikumpulkan ke node lain yang berdekatan. Data dikirimkan melalui transmisi radio akan diteruskan menuju BS (*Base Station*) yang merupakan penghubung antara node sensor dan user. Informasi tersebut dapat diakses melalui berbagai *platform* seperti koneksi internet atau satelit sehingga memungkinkan user untuk dapat mengakses secara *real time* melalui remote server.

Secara umum jaringan sensor nirkabel itu sendiri terdiri dari dua komponen, yaitu node sensor dan *sink*. Node sensor merupakan komponen kesatuan dari jaringan yang dapat menghasilkan informasi, biasanya merupakan sebuah sensor atau juga dapat berupa sebuah aktuator yang menghasilkan *feedback* pada keseluruhan operasi. Sink merupakan kesatuan yang mengumpulkan informasi dari node sensor sehingga dapat dilakukan pengolahan informasi lebih lanjut. Terdapat tiga bentuk sink yaitu *sink* dapat berupa node sensor yang lain dalam bentuk sensor dari jaringan itu sendiri atau dari jaringan lain. *Sink* dapat berupa sebuah komputer dan sebuah PDA yang digunakan untuk berinteraksi dengan Jaringan Sensor. Bahkan *sink* dapat berupa gateway ke jaringan yang lebih besar seperti internet sehingga interaksi dapat dilakukan melalui jarak yang sangat jauh dan tidak terkoneksi secara langsung dengan

Jaringan Sensor. Tugas utama dari sensor adalah memonitoring kondisi fisik dari suatu lingkungan dan menyampaikan hasilnya ke pusat kendali. Untuk data yang diperoleh bisa berupa pergerakan, tekanan atau temperatur.

Salah satu contoh WSN secara riil adalah digunakan pada *smart bridge* atau sistem SHM. Pada jembatan terdapat beberapa sensor node yang disebarakan pada beberapa titik jembatan seperti pada Gambar 2.3 (a) dan (b). Sensor node disini bertindak sebagai pendeteksi getaran dan temperatur. Jika terjadi masalah pada jembatan, seperti kabel yang terputus atau besi yang hilang, maka sebelum terjadi kecelakaan sensor node akan mengirimkan peringatan ke sink node atau base station seperti pada Gambar 2.3 (b).



a) WSN pada Long Span Bridge [24]

b) WSN pada Road Bridge [25]

Gambar 2.3. Contoh Aplikasi WSN

2.2.1. Karakteristik WSN

Pada WSN terdapat beberapa karakteristik yang diperlukan, diantaranya :

1. Jenis layanan *Type of Service*

Jenis layanan yang diberikan oleh jaringan komunikasi konvensional adalah bit bergerak dari satu tempat ke tempat lain.

2. *Quality of Service*

Erat kaitannya dengan jenis layanan jaringan adalah kualitas layanan tersebut. Kualitas layanan berhubungan dengan waktu pengiriman, yakni memperhatikan terhadap delay saat proses komunikasi data.

3. Skalabilitas

Skala WSN bervariasi, maka arsitektur dan protokol harus mampu bekerja dari skala kecil sampai skala besar.

4. Programmability

Node-node sensor dapat di program sesuai dengan keadaan lingkungan yang akan diamati.

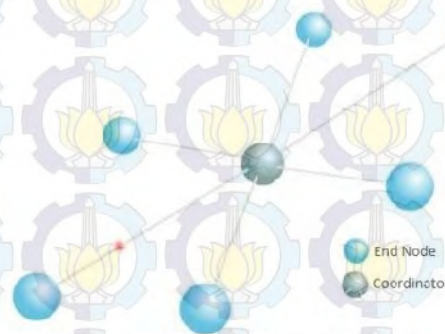
2.2.2. Topologi Jaringan WSN

Topologi jaringan adalah suatu cara untuk menghubungkan titik satu dengan titik lainnya. Masing-masing topologi mempunyai ciri khas, kelebihan dan kekurangan. Topologi yang di gunakan pada jaringan sensor nirkabel diantaranya:

1) Topologi *Star*

Topologi star adalah topologi paling dasar. Topologi star terdiri dari koordinator dan beberapa perangkat akhir (node). Dalam topologi ini, perangkat akhir berkomunikasi hanya dengan koordinator. Setiap pertukaran paket antara perangkat akhir harus melalui koordinator.

- Kerugian dari topologi ini adalah pengoperasian jaringan tergantung pada koordinator jaringan, dan karena semua paket antara perangkat harus melalui koordinator, koordinator mungkin menjadi bottlenecked. Juga, tidak ada jalan alternatif dari sumber ke tujuan.
- Keuntungan dari topologi star adalah paket melalui paling banyak dua hop untuk mencapai tujuan mereka.



Gambar 2.4 Topologi Star [16]

2) Topologi *Tree*

Topologi tree lebih kompleks dibandingkan dengan topologi star. Setiap node masih mempertahankan satu jalur komunikasi untuk *gateway*, perbedaannya menggunakan node-node lain dalam mengirimkan data, namun masih dalam satu jalur tersebut.

- Kelemahan untuk topologi ini adalah jika node router yang down, maka semua node yang bergantung pada node router akan kehilangan komunikasi ke gateway.

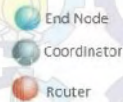


Gambar 2.5 Topologi Tree [16]

3) Topologi *Mesh*

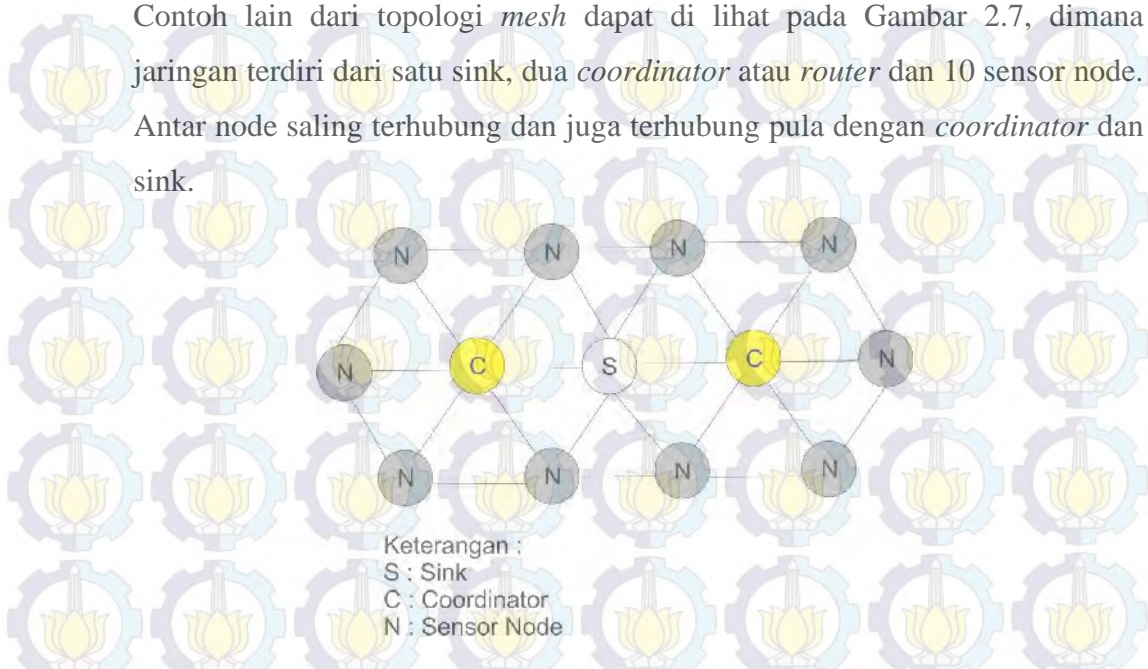
Topologi *Mesh* juga disebut sebagai jaringan peer-to-peer, terdiri dari satu koordinator, beberapa router, dan perangkat akhir. Topologi ini merupakan jalur komunikasi dimana masing-masing node dapat berkomunikasi dengan yang lainnya.

- Kelebihan dari topologi ini adalah dapat meningkatkan kehandalan sistem. Dalam sebuah jaringan *mesh*, node mempertahankan jalur komunikasi untuk kembali ke gateway, sehingga jika salah satu node router *down*, secara otomatis router data akan dilewatkan melalui jalur yang berbeda.
- Kelemahan pada topologi ini adalah adanya latensi atau delay karena data harus melalui beberapa hop sebelum mencapai *gateway*.

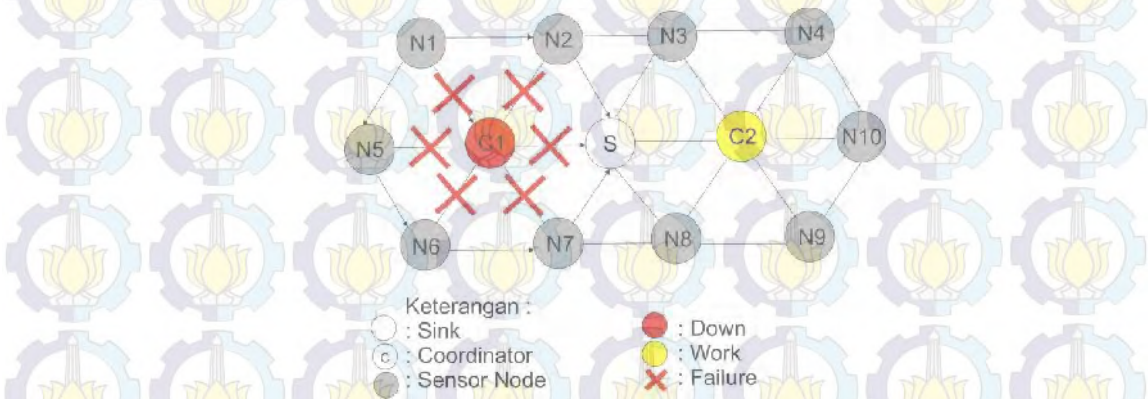


Gambar 2.6 Topologi *Mesh* [16]

Contoh lain dari topologi *mesh* dapat di lihat pada Gambar 2.7, dimana jaringan terdiri dari satu sink, dua *coordinator* atau *router* dan 10 sensor node. Antar node saling terhubung dan juga terhubung pula dengan *coordinator* dan sink.



Gambar 2.7. Topologi *Mesh* dengan 2 *coordinator* dan sink



Gambar 2.8. Topologi *Mesh* terdapat *coordinator* yang *down*

Melihat dari kelebihan topologi *mesh* yakni pada pernyataan bahwa topologi *mesh* mampu mempertahankan jalur komunikasi untuk kembali ke gateway, sehingga jika salah satu node router *down*, secara otomatis router data akan dilewatkan melalui jalur yang berbeda maka Seperti pada Gambar 2.8, ketika sensor node (N1) ingin mengirimkan data menuju sink melalui *coordinator* / *router* (C1). Namun, karena *coordinator* (C1) *down*, maka data tetap dapat dikirimkan menuju sink. Dengan topologi *mesh*, data dari N1 dapat dikirimkan menuju sink melalui N2 atau N5. Jika data diteruskan pada N2, maka data akan langsung dikirimkan menuju sink. Namun, jika data

diteruskan menuju N5, maka data akan diteruskan lagi ke N6, N7 dan terakhir menuju sink. Kelemahan pada topologi ini adalah adanya keterlambatan / delay karena data harus melalui beberapa hop sebelum mencapai *gateway* dimana pada Gambar 2.10 terdapat forward data dari N5, N6, N7, baru mencapai sink.

2.3. DSDV

DSDV merupakan kependekan dari *Destination Sequenced Distance-Vector* adalah algoritma routing protocol adhoc proaktif yang didasari pada Bellman – Ford yang pertama kali dikenalkan. Algoritma ini berkontribusi untuk mengatasi *Routing Loop*. Pada DSDV, digunakan *sequence number* untuk mengirimkan pesan pada jaringan. *Sequence number* dihasilkan juga saat ada perubahan dalam jaringan, hal ini terjadi karena sifat table routing node pada jaringan yang menggunakan protokol proaktif yang update secara periodik, serta *Trigered update* ulang digunakan oleh node untuk mengupdate node yang masuk dan keluar dari jaringan [10].

DSDV merupakan salah satu *Proactive Routing Protocol* yang memerlukan setiap node untuk mengirimkan tabel routing ke seluruh node tetangganya secara periodik. Metode routing DSDV yakni node yang berada dalam jaringan akan menjaga tabel routing ke node tetangganya. Tabel routing berisi alamat node tujuan, jumlah hop yang diperlukan agar sampai di tujuan dan *sequence number*. Saat tabel routing dalam satu node telah diupdate, maka node akan memilih rute untuk mencapai node tujuan dengan beberapa kriteria :

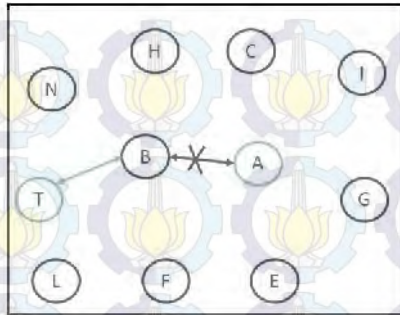
- Memiliki *sequence number* terbaru dengan melihat *sequence number* yang paling besar.
- Apabila nilai *sequence number* sama, maka akan dilihat nilai metricnya. Nilai *metric* yang dipilih adalah yang lebih kecil.

DSDV memiliki beberapa kelebihan, diantaranya [10]:

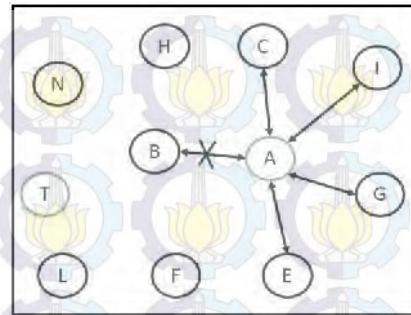
- DSDV menjamin tidak ada *looping route*
- DSDV dapat mereduksi masalah *count to infinity*
- DSDV dapat menghindari trafik lebih dengan kenaikan drastis update penuh untuk dump

- DSDV hanya memaintenance path terbaik menuju tujuan, dari sekian banyak path ketujuan.

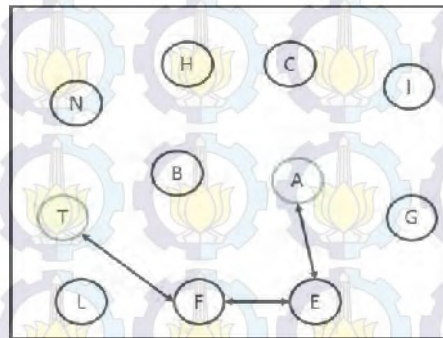
Ilustrasi pengiriman paket dengan routing protokol DSDV yakni ketika akan menyampaikan pesan dari node "A" untuk ke tujuan node "T" namun terdapat jalur yang breaks, maka protokol membuat link sementara melalui node tetangga yang memiliki rute valid menuju tujuan yang dikehendaki. Link sementara dibuat dengan mengirimkan satu-hop ROUTE-REQUEST dan pesan ROUTE-ACK. Node "A" ketika menemukan link hop yang rusak maka node akan menyebarkan (broadcast) satu hop paket ROUTE-REQUEST ke semua node tetangganya. Selanjutnya, node tetangga mengembalikan ROUTE-ACK didapatkan informasi bahwa terdapat rute untuk sampai ke tujuan namun node "A" bukan termasuk dalam rute tersebut. Setiap masukan pada tabel routing mempunyai masukan tambahan untuk waktu update rute. Waktu ini berada pada ROUTE-ACK yang digunakan dalam memilih rute sementara. Pada beberapa kasus terkadang ROUTE-ACK menerima update rute dengan nomor urut (sequence number) yang sama dan hop minimum, mengatasi hal ini maka node akan memilih rute dengan update rute terbaru. Pada Gambar 2.9 menunjukkan bagaimana node "A" membuat rute sementara untuk menuju node "T" ketika link diantaranya yakni dari node "A" ke node "B" breaks. Disini node "A" menunda terlebih dahulu pengiriman pakatnya sebagaimana ditunjukkan pada Gambar 2.9a. Selanjutnya node "A" menyebarkan (broadcast) paket ROUTE-REQUEST kepada node tetangganya. Node C, E, dan G merespon dengan mengirimkan paket ROUTE-ACK kembali waktu update rute dan hop ke node "A" seperti di tunjukkan pada Gambar 2.9b. Tabel 2.1 menunjukkan informasi update rute yang di terima oleh node "A". Dari data yang diperoleh pada tabel dapat dilihat bahwa node "C" dan "E" memiliki nilai yang sama untuk jumlah hop yang dapat dilewati, akan tetapi waktu update rute pada node "E" lebih besar dibandingkan node "C", yang berarti rute melalui E adalah update rute terbaru. Sehingga untuk pengiriman paket dari node "A" menuju node "T", node "A" memilih node "E" sebagai node selanjutnya seperti terlihat pada Gambar 2.9c.



a) Link Dari Node A ke B Breaks



b) Node A Menyebarkan ROUTE-REQUEST ke Node Tetangga



c) Node A Memilih Node E Sebagai Hop Selanjutnya

Gambar 2.9 Pembuatan Rute Sementara Node A

Tabel 2.1 Update Rute pada Node A

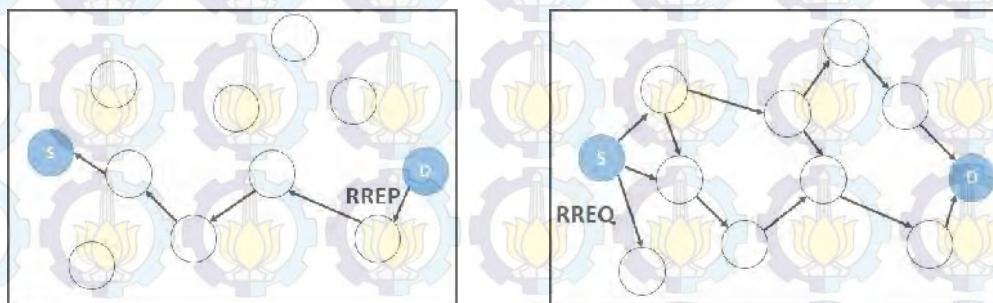
Node Tetangga	Jumlah Hop	Node Selanjutnya	Waktu Update (ms)
C	2	H	1765
E	2	F	1860
G	3	E	1050

2.4. AODV

Ad hoc On-demand Distance Vector (AODV). AODV adalah distance vector routing protocol yang termasuk dalam klasifikasi reaktif routing protocol, yang hanya me-request sebuah rute saat dibutuhkan. AODV yang standar ini dikembangkan oleh C. E. Perkins, E.M. Belding-Royer dan S. Das pada RFC 3561 [12].

Karena AODV merupakan on-demand, sebuah rute dibangun hanya jika dibutuhkan oleh node sumber untuk mentransmisikan paket data dan AODV menjaganya selama rute ini dibutuhkan. AODV menggunakan *sequence number* yang dibuat node tujuan untuk menentukan jalur terbaru ke node tujuan. Sebuah node akan memperbarui informasi rute hanya jika *sequence number* tujuan paket yang diterima sekarang lebih besar dibandingkan *sequence number* yang disimpan pada node. Hal ini mengindikasikan barunya rute yang diterima oleh node sumber. Untuk menghindari pengiriman ganda pada paket yang sama, AODV menggunakan nomer identitas pengiriman yang menjamin bebas *looping* karena node intermediate hanya meneruskan salinan pertama dari paket yang sama dan membuang duplikasi salinan.

Untuk menentukan rute ke node tujuan, node sumber mengirimkan paket *Route Request* (RREQ) ke jaringan. RREQ berisi identitas node sumber, identitas node tujuan, nomer urut tujuan, nomer urut sumber, identitas pengiriman dan TTL (*Time to Live*). Node yang menerima RREQ baik itu node tujuan atau tidak, node tersebut memiliki rute terbaru ke node tujuan dapat merespon ke RREQ untuk mengirimkan paket *Route Reply* (RREP) kembali ke node sumber.



a) Node sumber S mengirim RREQ ke node tetangganya

b) Node tujuan mengirimkan RREP kembali ke sumber

Gambar 2.10 Mekanisme Penemuan Rute

Ketika sebuah node meneruskan paket RREQ ke node tetangganya, node tersebut juga mencatatnya pada tabel routing dari node mana salinan RREQ itu datang. Informasi ini digunakan untuk membangun reverse path untuk paket RREP. AODV hanya menggunakan link yang simetris karena paket RREP mengikuti reverse path dari paket RREQ. Ketika sebuah node menerima sebuah paket RREP, informasi tentang node sebelumnya darimana paket RREP tersebut

di terima juga disimpan dengan tujuan untuk meneruskan paket data ke node berikutnya sebagai hop berikutnya menuju tujuan. ketika node sumber menerima sebuah paket RREP, rute tersebut sudah siap digunakan untuk mengirimkan data.

Selanjutnya node sumber mengirimkan kembali paket RREQ jika tidak menerima paket RREP sebelum kadaluarsa. Node sumber melakukan pencarian rute dengan usaha yang maksimal. Jika node sumber berpindah maka node sumber bisa memulai kembali pencarian rute node tujuan [9]. Mekanisme penemuan rute ditunjukkan pada Gambar 2.10.

2.5. Zigbee (802.15.4)

Zigbee adalah salah satu protokol pada jaringan nirkabel yang didesain oleh *zigbee* Alliance. *Zigbee* juga merupakan protokol yang telah dirancang khusus untuk komunikasi jarak pendek di WSN. Semua lapisan pada *zigbee* didasarkan pada standar IEEE 802.15.4. Lapisan pada *zigbee* meliputi lapisan fisik, lapisan jaringan, lapisan aplikasi dan lapisan keamanan. *Zigbee* memiliki keunggulan pada bentuknya yang minimalis dan pengoperasiannya yang mudah. *Zigbee* didesain untuk melakukan komunikasi jarak pendek, yaitu dengan jarak komunikasi hanya sekitar 50 meter hingga 100 meter. Sedangkan kecepatan komunikasi yang dapat dilakukan *zigbee* hanyalah 250 kbps. Terdapat perbedaan jika kecepatan *zigbee* dibandingkan dengan sistem komunikasi jarak pendek lainnya, misalnya Wi-Fi yang memiliki kecepatan komunikasi hingga 54 Mbps.

2.5.1. Karakteristik ZigBee

ZigBee memiliki karakteristik diantaranya :

- Bekerja pada tiga rentang frekuensi yakni frekuensi 2.4 GHz, 868MHz dan 915MHz. Frekuensi 868-870 MHz dengan 1 kanal, frekuensi 902-928 MHz dengan 10 kanal dan frekuensi 2,4 GHz dengan 16 kanal (digunakan di Indonesia).
- Mempunyai konsumsi daya yang rendah
- Maksimum transfer rate untuk tiap data pada tiap lebar pita adalah sebagai berikut 250Kbps untuk 2.4GHz, 40 kbps untuk 915 MHz, dan 20Kbps untuk 868 MHz

- Mempunyai Throughput yang tinggi dan dan latency yang rendah untuk duty cycle yang kecil.



Gambar 2.11 Perangkat Zigbee

2.5.2. Arsitektur ZigBee

Arsitektur lapisan komunikasi pada zigbee berdasarkan standar IEEE 802.15.4 terdapat beberapa lapisan yang terdiri dari lapisan berikut ini :

1. Physical Layer

Physical Layer (Layer Fisik) merupakan komponen yang sangat penting dalam proses komunikasi antar perangkat. Layer ini biasanya digunakan untuk mengubah data logika menjadi bentuk yang sesuai untuk dikirimkan pada media transmisi yang digunakan. Juga sebagai interface (antarmuka) dan penentu kualitas dari sebuah komunikasi, sekaligus bertugas mensuplai berbagai macam informasi dari layer di atasnya.

2. Media Access Control (MAC) Layer

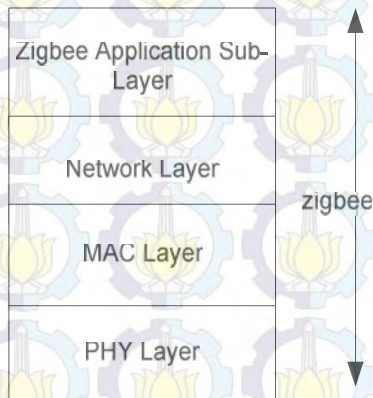
Layer ini digunakan untuk mengakses saluran yang digunakan. Dimana terdapat dua mekanisme untuk mengakses saluran, yaitu mode Beacon yang menggunakan teknik CSMA/CA dan mode non-Beacon yang menggunakan teknik non CSMA/CA.

3. Network Layer.

Network Layer digunakan untuk mengatur jaringan, antara lain : konfigurasi perangkat, pengalamatan, penggabungan jaringan, sistem keamanan jaringan.

4. Application Layer

Application Layer digunakan untuk mencocokkan antar dua perangkat yang berkomunikasi dalam waktu bersamaan dan menyampaikan pesan antar dua perangkat tersebut.



Gambar 2.12. Arsitektur ZigBee

2.5.2. XBee Pro versi v1

XBee Pro versi v1 merupakan salah satu produk dari Digi International, Inc. yang mendukung dan memenuhi standar IEEE 802.15.4 dan dirancang untuk komunikasi tanpa kabel dengan band frekuensi 2,4 GHz. Salah satu keunggulan XBee Pro ini adalah konsumsi daya yang sangat rendah, sehingga memiliki lifetime yang sangat lama. Pada XBee Pro versi v1 memiliki spesifikasi sebagai berikut :

XBee Pro versi v1 digunakan dalam komunikasi data antar sensor node maupun komunikasi data antara sensor node dengan sink. Pada *transceiver* XBee modul mempunyai dua mode operasi :

- 1) Transparent serial port mode. Pengiriman data dari sensor ke modul XBee melalui serial port, kemudian XBee module mengirimkan data ke module XBee lainnya secara *wireless*.
- 2) Packet mode. Pengiriman pesan ke module XBee itu sendiri. Terdapat dua macam *packet* mode, yaitu IO packet dan command packet.

Tabel 2.2 Spesifikasi Modul XBee Pro versi v1 [13]

No	Spesifikasi	Nilai
1)	Transmit Power Output	<ul style="list-style-type: none">• 63 mW (18 dBm) pada wilayah Eropa• 10 mW (10 dBm) untuk international
2)	Receiver Sensitivity	-100 dBm
3)	RF Data Rate	250.000 bps
4)	Supply Voltage	2,8 – 3,4 V
5)	Transmit Current	250 mA (@3,3 V)
6)	Receive Current	55 mA (@3,3 V)
7)	Generating Frequency	ISM 2,4 GHz
8)	Antenna Type	Omni-directional Gain : 2,1 dBi

RF Modul Operation. Komunikasi XBee/ XBee-Pro ke host device adalah melalui sebuah logic-level asynchronous serial port. Melalui serial port ini, modul dapat berkomunikasi dengan beberapa logic atau voltage compatible UART, atau melalui sebuah level translator ke beberapa perangkat serial (misalnya: melalui RS 232 atau USB interface board).

2.6. Arduino Due

Arduino Due diperkenalkan pada tahun 2005 dan dirancang dengan harga yang murah dan didesain untuk penggunaan sensor dan aktuator. Arduino Due merupakan board mikrokontroler yang berbasis CPU Atmel SAM3X8E ARM Cortex - M3. Arduino due ini merupakan board pertama dengan 32-bit ARM core microcontroller yang memiliki 54 pin input/output digital (12 pin digunakan untuk output PWM), 12 input analog, 4 UART (serial port hardware), 84 MHz clock, koneksi USB OTG, 2 DAC (Digital to Analog), 2 TWI, power jack, SPI header, JTAG header, dan tombol *reset* dan *delete* [26]. Sedangkan Tabel 2.2 merupakan ringkasan dari spesifikasi Arduino Due.



Gambar 2.13. Arduino Due

Tabel 2.3. Spesifikasi Arduino Due

No.	Parameter	Nilai
1)	Mikrokontroler	AT91SAM3X8E
2)	Operating Voltage	3,3 V
3)	Input Voltage (recommended)	7 – 12 V
4)	Input Voltage (limits)	6 – 16 V
5)	Digital I/O Pins	54 (of which 12 provide PWM output)
6)	Analog Input Pins	12
7)	Analog Output Pins	2 (DAC)
8)	Total DC Output Current on all I/O lines	130 mA
9)	DC Current for 3,3 V Pins	800 mA
10)	DC Current for 5 V Pins	800 mA
11)	Flash Memory	512 KB all available for the user applications
12)	SRAM	96 KB (two banks : 64 KB and 32KB)
13)	Clock Speed	84 Hz

2.7. Sensor

Sensor secara umum merupakan alat untuk mendeteksi atau mengukur besaran fisis seperti panas, sinar, kimia, getaran dan lain-lain dan mengubahnya menjadi sinyal elektrik seperti arus listrik maupun tegangan. Sensor pada SMKS

yang umumnya diimplementasikan terdapat beberapa macam diantaranya *accelerometer, strain gauges, temperature sensor, displacement transducers, level sensing station, anemometer, dynamic weight-inmotion sensor*. Namun, pada penelitian ini hanya menfokuskan pada sensor accelerometer saja.

2.7.1 Sensor Accelerometer

Sensor accelerometer merupakan sensor yang digunakan untuk mengukur percepatan, mendeteksi dan mengukur getaran (vibrasi), dan mengukur percepatan akibat gravitasi (inklinasi). Sensor accelerometer dibutuhkan untuk mengukur getaran yang terjadi pada jembatan dalam implementasi sistem SHM jembatan.

Biasanya, jembatan mengalami getaran saat terdapat beban atau kendaraan yang melewati jembatan. Pada [26], sensor accelerometer yang digunakan dengan tipe MMA 7361 dengan triple axis dimana accelerometer akan mendeteksi getaran dengan arah sumbu-X, sumbu-Y dan sumbu-Z.

2.8. Standar IEEE 802.15.4

Standar IEEE 802.15.4 bertujuan menghasilkan harga yang murah dan energi komunikasi yang kecil. Pada standar IEEE 802.15.4 tahun 2003, ditetapkan dua pilihan *physical layer* di pita frekuensi berbeda namun MAC bekerja efektif.

Pada tahun 2006, terdapat tambahan beberapa pilihan *physical layer*, sedangkan frame pada MAC mengalami peningkatan versi, peningkatan keamanan, dan didukung basis *shared-time* dengan metode *stamping time*, penjadwalan beacon dan sinkronisasi pesan *broadcast beacon* pada *beacon enabled* [25].

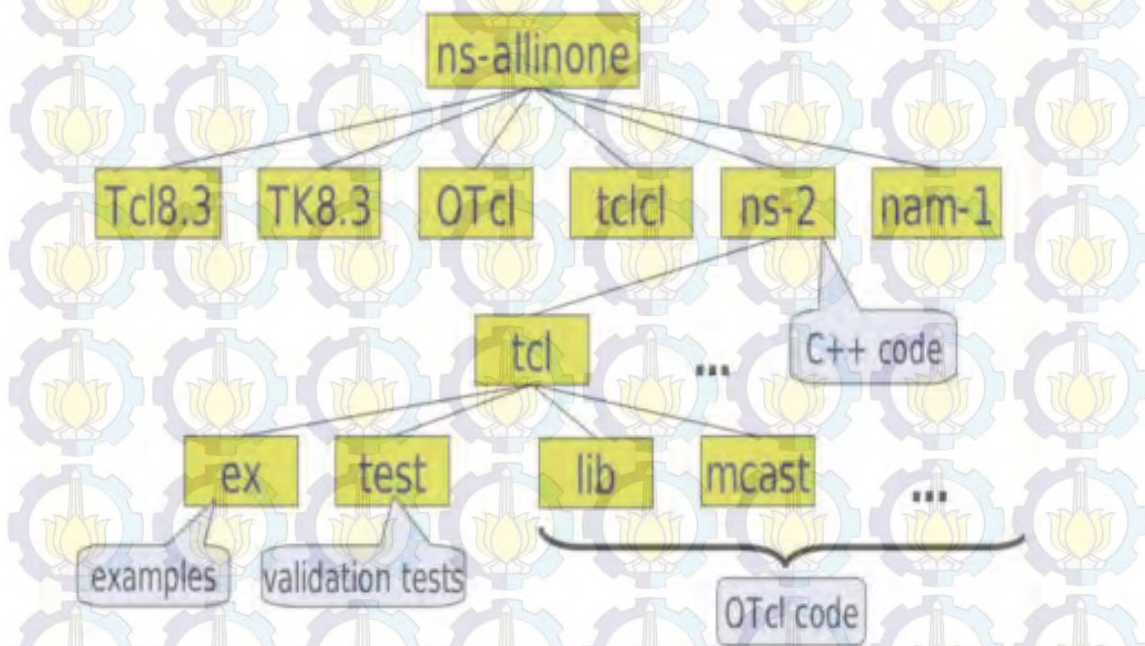
2.9. Network Simulator-2 (NS-2)

Network Simulator (NS) pertama kali dibangun sebagai varian dari REAL Network Simulator pada tahun 1989 di UCB (University of California Berkeley). NS2 merupakan suatu sistem yang bekerja pada OS Unix/Linux namun juga bisa dijalankan pada OS Windows namun harus menggunakan Cygwin Linux *Enviromentnya*. NS2 dibangun dari 2 bahasa pemrograman yaitu C++ dan OTcl, C++ sebagai *library* yang berisi *event scheduler*, protokol, dan komponen

jaringan yang diimplementasikan pada simulasi oleh user . Sedangkan OTcl digunakan pada script simulasi yang ditulis oleh NS user .Otcl juga berperan sebagai interpreter. Bahasa C++ digunakan pada library karena C++ mampu mendukung runtime simulasi yang cepat, meskipun simulasi melibatkan simulasi jumlah paket dan sumber data dalam jumlah besar. Sedangkan bahasa Tcl memberikan respon runtime yang lebih lambat daripada C++, namun jika terdapat kesalahan, respon Tcl terhadap kesalahan syntax dan perubahan script berlangsung dengan cepat dan interaktif.

2.9.1. Komponen Pembangun NS2

Untuk komponen pembangun NS2 dapat dilihat pada Gambar 2.11 berikut :



Gambar 2.14 Komponen Pembangun NS2

Keterangan:

- Tcl : Tool command language
- OTcl : Object Tcl
- TK : Tool Kit
- Tclcl : Tcl/C++ Interface
- NS2 : NS versi 2
- Nam : Network animator

Pada Gambar 2.11 terlihat bahwa komponen pembangun NS2 yakni pada file folder ns-allinone terdapat beberapa *tool* yakni Tcl8.3 sebagai *tool command*, Tk8.3 sebagai *tool kit*, Otcl sebagai *object tcl*, Tctl sebagai *interface*, NS2 sebagai simulatornya dan nam sebagai GUI yang menampilkan hasil simulasi. Pada *tools* Tctl terdapat file tcl dan C++ yang berisi program atau kode-kode. Pada Tcl terdapat beberapa file di dalamnya yakni Ex yang berisi contoh-contoh program yang sudah ada, Test untuk validasi, Lib yang merupakan *library* dan Mcast. Lib dan Mcast termasuk pada pengkodean Otcl. Tcl adalah bahasa pemrograman yang didasarkan pada string-string based command. Tcl di desain untuk menjadi perekat dalam membangun software *building block* untuk menjadi suatu aplikasi. Sedangkan Otcl adalah ekstensi tambahan pada Tcl yang memungkinkan fungsi *object oriented*.



(HALAMAN INI SENGAJA DIKOSONGKAN)

BAB 3 METODE PENELITIAN

3.1. Rancangan Penelitian

Sistem monitoring kesehatan struktur sudah diterapkan pada berbagai objek bangunan, dengan *constrain* yang berbeda menjadikan sistem dalam monitoring pun berbeda mengikuti objek yang diamati. Penelitian ini berfokus pada jembatan bentang panjang Suramadu yakni jembatan yang menghubungkan antara pulau Jawa dan Madura. dalam penelitian ada banyak hal yang perlu diperhatikan dari jembatan seperti kondisi lingkungan dan struktur. Penelitian ini merupakan penelitian besar. Beberapa penelitian yang telah dilakukan pada Jembatan Suramadu ini dapat dilihat pada road map pada Gambar 3.1 di bawah ini.

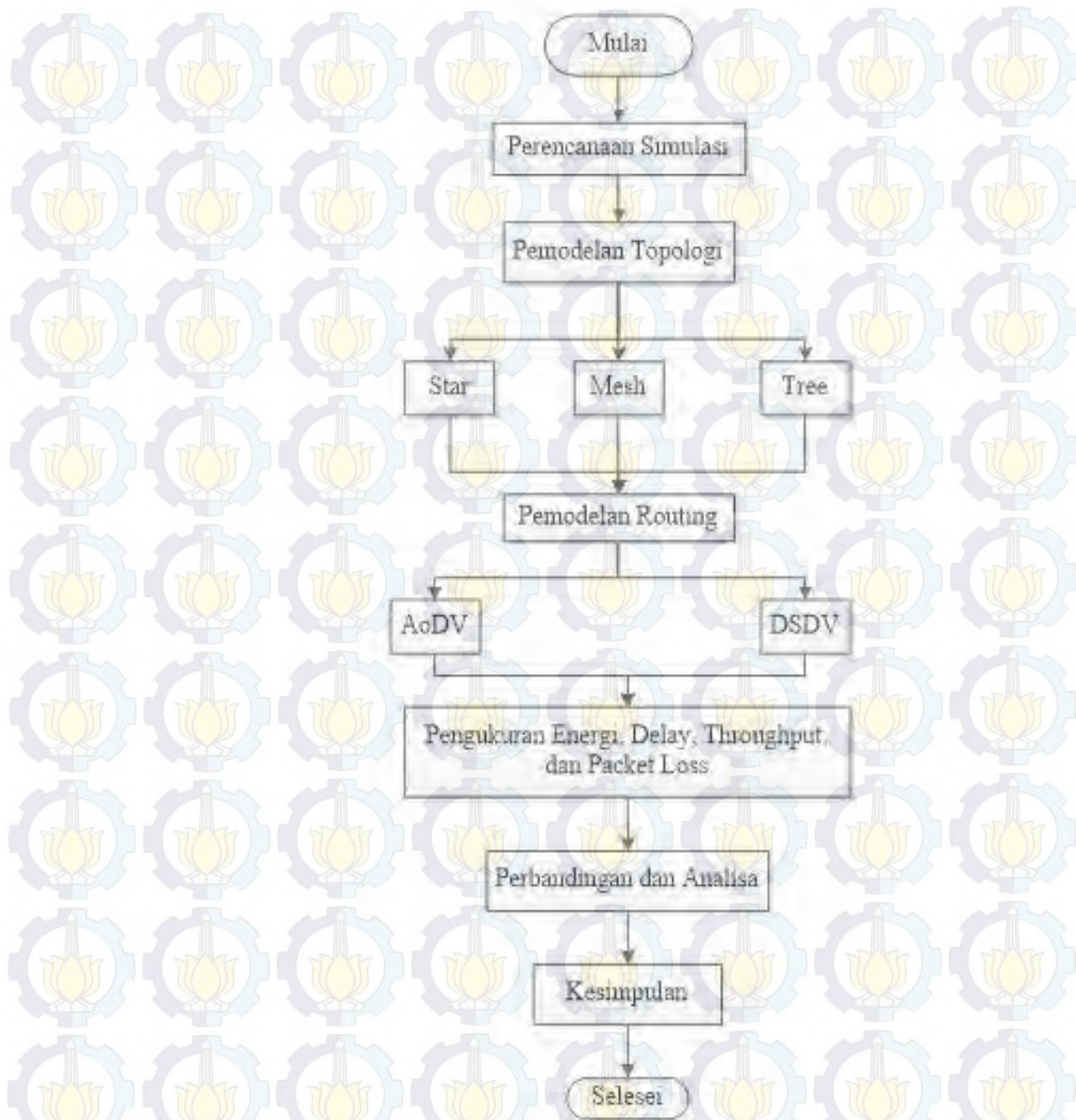


Gambar 3.1 Road Map Penelitian

Tahap awal penelitian terdapat beberapa fokus penelitian yang telah dilakukan. Pertama yakni melakukan penelitian tentang sistem akuisisi data pada sistem monitoring kesehatan struktur Jembatan Suramadu, dengan memastikan data sensor dapat diterima oleh server. Kedua yakni penelitian tentang pemantauan secara terpusat pengiriman data sensor ke server yang di fokuskan tidak hanya pada satu jembatan saja, melainkan beberapa jembatan dapat dipantau secara terpusat. Ketiga melakukan simulasi jaringan untuk mengetahui topologi dan desain routing terbaik dan cocok untuk diterapkan pada sistem monitoring kesehatan struktur jembatan[17]. Mengacu dari tahap awal penelitian, belum adanya pengaturan energi yang digunakan. Manajemen energi berperan penting

dalam jaringan sensor nirkabel, agar penggunaan energi lebih efisien maka perlu adanya teknik untuk manajemen energi yang digunakan. Karena menggunakan teknologi nirkabel tanpa adanya kabel yang terpasang ke jala-jala PLN membuat jaringan sensor nirkabel menggunakan baterai sebagai sumber daya. Keberadaan sumber daya ini memiliki *durability* yang terbatas dan perlu adanya pergantian. Maka pada tahap kedua dilakukan manajemen energi pada konsumsi daya dan sumber dayanya. Pada tahap kedua ini dilakukan penelitian untuk manajemen energi dengan cara mengatur daya yang dikonsumsi oleh jaringan sensor nirkabel seminimal mungkin menggunakan routing dan topologi yang tepat. Seperti ketika node sensor belum melebihi *threshold* maka berstatus *idle* sehingga daya yang dikonsumsi tidak sebanyak ketika pengiriman data ataupun ketika node sensor aktif. Namun pada penelitian tahap pertama pada bagian ketiga [17] masih terdapat kekurangan yakni untuk desain peletakan sensor jembatan yang digunakan belum memenuhi gambaran Jembatan Suramadu secara *real time*. Maka pada penelitian ini dilakukan desain ulang peletakan sensor dengan mengacu pada Jembatan Suramadu tersebut. Sebelum melakukan simulasi dari penelitian, dirancang terlebih dahulu alur penelitian yang akan dilakukan. Tahapan-tahapan yang akan dilakukan dapat dilihat pada Gambar 3.2.

Penelitian dimulai dengan perencanaan simulasi meliputi pemodelan sistem yang akan digunakan diantaranya jenis antrian, banyaknya paket data dan banyak node, selanjutnya pemodelan topologi memanfaatkan topologi star, tree dan mesh dengan menggunakan karakteristik radio module berupa Zigbee (802.15.4). Setelah pemodelan topologi, dilakukan pemodelan routing protokol. Terdapat dua macam routing protokol yang akan digunakan yakni AODV dan DSDV. Pada setiap routing protokol dilakukan pengukuran beberapa parameter yakni energi, *delay*, *throughput* dan *packet loss*. Setelah di dapat hasil dari parameter yang diukur, selanjutnya menganalisa hasil pengukuran dan kinerja dari routing protokol yang kemudian kedua hasil tersebut di bandingkan untuk di dapatkan kesimpulan.



Gambar 3.2 Flowchart Rencana Penelitian

Untuk proses manajemen energi dari node sensor, dilakukan dengan cara mengatur desain topologi dan routing yang digunakan.

3.2. Manajemen Sumber Daya

Sebuah sistem pada jaringan sensor nirkabel memiliki kemandirian untuk sumber daya yang dikonsumsi. Namun terdapat permasalahan penting yakni dalam proses bagaimana energi yang terbatas pada jaringan sensor nirkabel ini

dapat memberikan daya yang optimal. Pada penelitian [26] penghematan konsumsi daya dilakukan dengan cara mengurangi jumlah data pengiriman, penjadwalan pemancar radio dan komponen sensor, penambahan interval waktu pengiriman data dan deteksi sensor. Dengan menggunakan penelitian sebelumnya sebagai acuan dalam penentuan konsumsi daya yang digunakan oleh jaringan sensor nirkabel maka dapat diketahui berapa daya yang harus dicover. Penggunaan energi yang dikonsumsi digolongkan dalam beberapa macam, yakni :

1. Menghindari dan menghemat aktifitas yang membuang energi.
2. Mengestimasi penggunaan energi pada semua subsistem dan akibatnya pada *lifetime* jaringan sensor nirkabel.

Sedangkan konsumsi daya yang diperlukan oleh sistem, yakni :

Tabel 3.1 Konsumsi Daya pada Sistem

Karakteristik	Spesifikasi	Daya (Volt)
Sensing	ADXL345	2 - 3.6
	Strain gage	1
Komputing	Arduino due SAMX8E	3.3 - 5
Komunikasi	Xbee pro	2.8 - 3.4
Total Daya		13 Volt

Dengan mengacu pada Tabel 3.1, maka pada penelitian ini dilakukan metode untuk penghematan energi dengan menggunakan metode pengaturan routing dan topologi yang sesuai dengan desain jembatan. Teknik manajemen konsumsi daya didapatkan melalui pengaturan node-node sensor dengan memanfaatkan kondisi *sleep/awake* pada node sensor tersebut. Node *awake* ketika terjadi proses pentransmisian data dengan node tetangganya. Sedangkan untuk node yang tidak mentransmisikan data dalam kondisi *sleep*.

3.3. Metode Pengiriman Data

Pada proses pengiriman data perlu diperhatikan konsumsi daya yang diperlukan oleh node. Selain proses manajemen energi sumber daya, perlu adanya suatu metode untuk memajemen konsumsi dayanya. Suatu node apabila berada pada kondisi selalu aktif maka dapat menghabiskan *supply* daya yang ada. Sehingga perlu adanya suatu metode untuk menghemat konsumsi daya tersebut. Pada penelitian ini ditawarkan dengan model komunikasi berdasarkan terjadinya

suatu kejadian. Node yang telah dibagi dalam 2 macam yakni node sensor dan node koordinator/*sink*. Node sensor memiliki fungsi sebagai sensing node yang akan mensensing adanya getaran yang tertangkap di daerah sensing node. Node sensor yang berada paling ujung (*End device*) melanjutkan pengiriman datanya melalui node sensor terdekatnya. Node sensor yang berada dekat dengan node koordinator (router) akan menerima data dari node sensor sebelumnya dan data hasil pembacaan sensor dikirimkan pada node koordinator. Sedangkan node koordinator berfungsi sebagai pengumpul data sensing dari node sensor yang kemudian data akan dikirimkan pada base station. Berikut metode pengiriman data pada WSN :

1. Node sensor berada pada kondisi *stand by* untuk melakukan proses penginderaan getaran.
2. Ketika ada mobil melintas dan mulai memasuki area node sensor dan berada pada posisi node sensor maka node sensor mulai melakukan penginderaan terhadap getaran.
3. Setelah keluar dari area node maka data akan dikirim ke node koordinator.
4. Kemudian data di teruskan ke base station.

Disini terdapat banyak node sensor yang diletakkan berseberangan, ketika node sensor sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *awake* (node aktif), sedangkan ketika node sensor tidak sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *sleep*. *Sleep* bukan berarti node mati namun node berada pada kondisi *idle* atau hanya mendengar saja. Metode ini mampu menghemat daya yang dibutuhkan oleh node. Karena pada kondisi *awake* dan *sleep* lebih banyak membutuhkan daya pada kondisi *awake*.

3.4. Pemodelan Sistem

Pada penelitian ini dilakukan simulasi dengan menggunakan sistem WSN dengan membuat node sederhana menggunakan parameter dibawah ini:

Jenis kanal	: kanal Wireless
Jenis propagasi	: propagasi TwoRayGround
Jenis layer fisik	: 802.15.4
Jenis layer MAC	: 802.15.4

Jenis antrian : Queue/DropTail/PriQueue

Jenis link layer : LL

Jenis antenna : antena Omni

Maksimal paket pada antrian : 150

Tinggi Antenna : 0,6682 m

Loss : 1 dB (tidak terjadi attenuasi)

Gain Antena : 2,1 dBi

Berdasarkan penelitian [17] proses simulasi perlu diketahui beberapa parameter lain yakni : transmit power output, receive threshold, transmit power, receive power dan initial energy.

3.4.1. Transmit Power Output (Pt)

Transmit power Output (Pt) adalah daya sinyal yang ditransmisikan. Daya sinyal yang dikeluarkan oleh transmitter agar pesan informasi dari transmitter dapat diterima oleh receiver. Disini berdasarkan datasheet Xbee pro transmit power output yang digunakan adalah 60 mW (18 dBm) atau 0,06 W.

3.4.2. Receive Threshold (RxThresh)

Receive Threshold adalah sinyal daya terkecil yang digunakan untuk mendeteksi dan menerima paket yang telah ditransmisikan. Receive threshold menjadikan receiver hanya akan dapat menerima sinyal yang dikirimkan oleh transmitter dengan daya sinyal yang lebih besar dari receive threshold. Disini berdasarkan datasheet Xbee pro memiliki receiver sensitivity sebesar -100 dBm sehingga jika diubah menjadi satuan Watt adalah :

$$10 \times \log_{10} (\text{RxThresh}/1 \text{ mW}) = -100$$

$$\log_{10} (\text{RxThresh}/1 \text{ mW}) = -10$$

$$\text{RxThresh}/1\text{mW} = 10^{-10}$$

$$\text{RxThresh} = 10^{-10} \times 1 \text{ mW}$$

$$= 10^{-13} \text{ W}$$

3.4.3. Transmit Power (Tx)

Pada waktu sebuah transmitter akan mengirimkan sinyal informasi menuju receiver, transmitter tersebut akan mengeluarkan daya untuk melakukan transmisi.

Transmit power adalah daya yang dibutuhkan oleh transmitter untuk mentransmisikan sinyal informasi tersebut. Transmit power dapat dituliskan dengan persamaan sebagai berikut.

$$txPower = V_t \times I_t \quad (3.1)$$

Keterangan :

$txPower$ = Transmit Power (dalam Watt)

V_t = Tegangan yang digunakan dalam transmisi (dalam Volt)

I_t = Arus yang digunakan dalam transmisi (dalam Ampere)

Berdasarkan datasheet Xbee pro tegangan yang digunakan bernilai 3,3 V dan transmit current bernilai 250 mA. Maka transmit power yang dihasilkan adalah :

$$\begin{aligned} txPower &= V_t \times I_t \\ &= 3,3 \times 0,25 \\ &= 0,825 \text{ W} \end{aligned}$$

3.4.4. Receive Power (Rx)

Pada waktu sebuah receiver menerima sinyal dari informasi dari transmitter maka terdapat daya yang harus dikeluarkan sebuah receiver tersebut. Maka receive power adalah daya yang dibutuhkan oleh receiver untuk menerima sinyal informasi yang dikirimkan oleh transmitter. Receive power dapat ditulis dengan persamaan berikut.

$$rxPower = V_r \times I_r \quad (3.2)$$

Keterangan :

$rxPower$ = Receive Power (dalam Watt)

V_r = Tegangan yang digunakan dalam penerimaan (dalam Volt)

I_r = Arus yang digunakan dalam penerimaan (dalam Ampere)

Berdasarkan datasheet Xbee pro digunakan tegangan yang bernilai 3,3 V dan receive current bernilai 55 mA. Maka transmit power yang dihasilkan adalah :

$$\begin{aligned}rxPower &= V_r \times I_r \\ &= 3,3 \times 0,055 \\ &= 0,18 \text{ W}\end{aligned}$$

3.4.5. Initial Energy (E)

Initial energy adalah energi yang digunakan dalam perangkat. Energi yang dimaksud adalah besarnya energi yang akan digunakan dalam sebuah perangkat komunikasi. Besarnya energi yang digunakan dalam perangkat adalah bergantung dari tegangan dan arus dari spesifikasi perangkat serta waktu yang dibutuhkan untuk melakukan komunikasi. Intial energy dapat dibuat persamaan yaitu :

$$E = V \times I \times t \quad (3.3)$$

Keterangan :

E = Initial Energy (Joule)

V = Tegangan sumber energi (Volt)

I = Arus dari konstan discharge (Ampere)

t = Waktu (sekon atau detik)

Pada sistem SMKS dalam penelitian ini, baterai yang digunakan adalah baterai *brand* Imedion berkapasitas 9.6 V dan 220 mA untuk 24 jam [26].

Sehingga initial energy-nya adalah :

$$\begin{aligned}E &= V \times I \times t \\ &= 9.6 \times 0,024 \times 24 \times 3600 \\ &= 19906 \text{ J}\end{aligned}$$

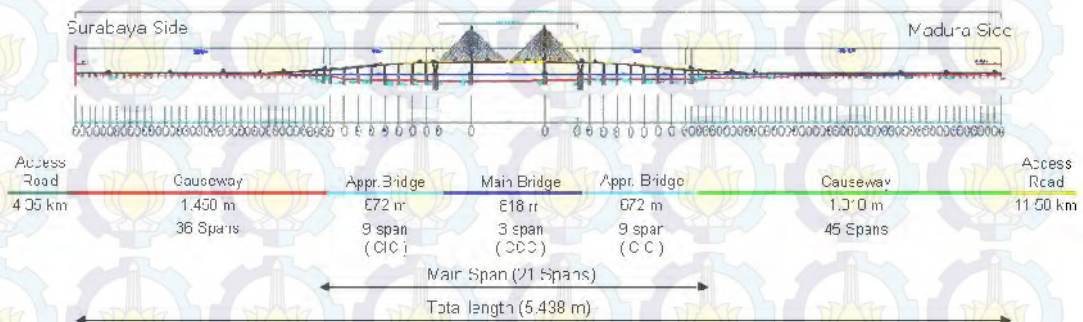
3.5. Desain Sensor pada Jembatan

Dalam peletakan titik-titik sensor pada jembatan harus mengetahui terlebih dahulu bagaimana gambaran jembatan dan kondisi jembatan tersebut. Desain

jembatan dalam simulasi didasarkan pada bentuk asli Jembatan Suramadu dengan memperlihatkan dari sisi samping dan atas.

3.5.1. Jembatan Suramadu

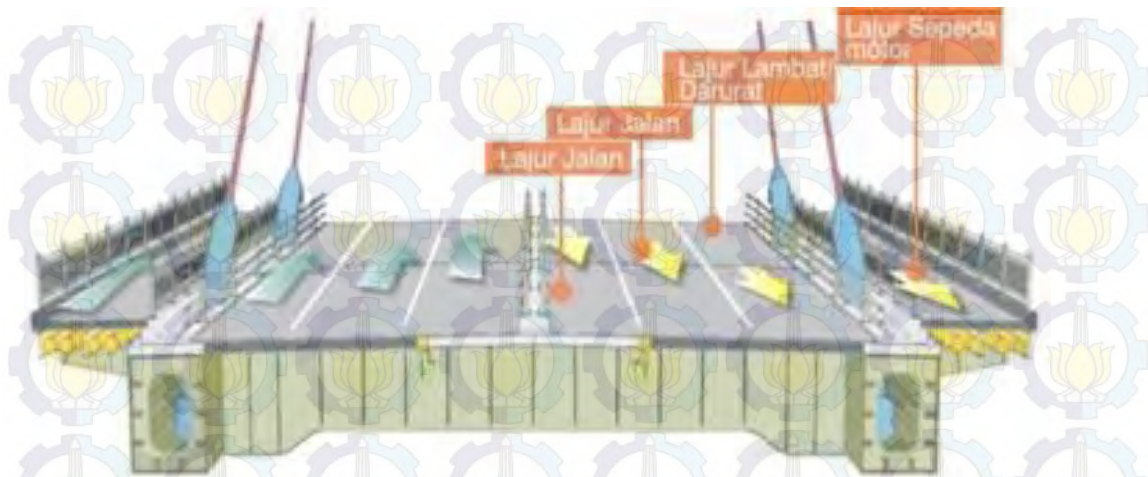
Jembatan Suramadu adalah jembatan yang menghubungkan antara pulau Jawa dan Madura. Pada dasarnya Jembatan Suramadu merupakan gabungan dari tiga jenis jembatan yakni jalan layang atau *causeway*, jembatan penghubung atau *approach bridge* dan jembatan utama atau *main bridge* yang memiliki panjang keseluruhan sepanjang 5.438 meter dengan lebar kurang lebih 30 meter.



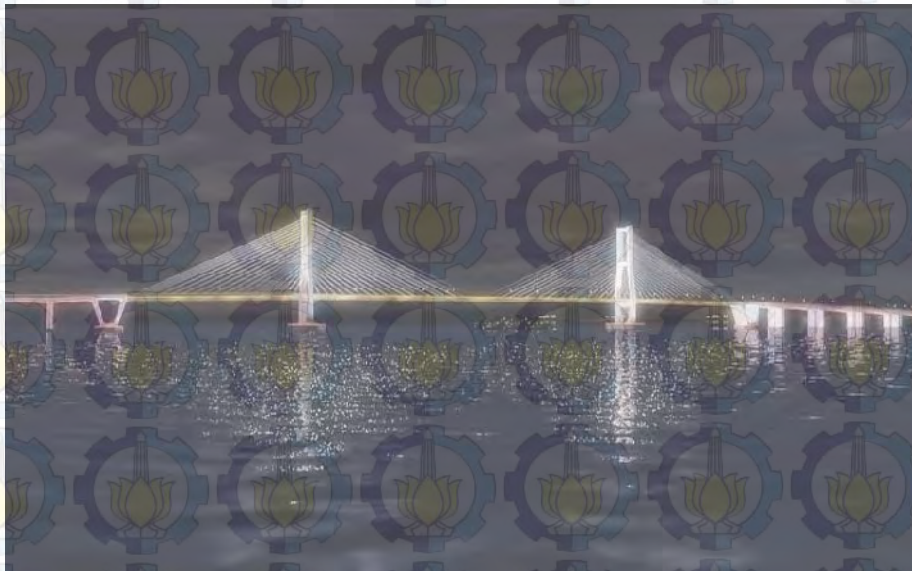
Gambar 3.3 Struktur Jembatan Suramadu

Jembatan ini memiliki spesifikasi pembagian lajur jalan berdasarkan lebar yakni sebagai berikut :

- Lajur jalan utama sebanyak 2 lajur
- Lajur lambat (darurat)
- Lajur sepeda motor



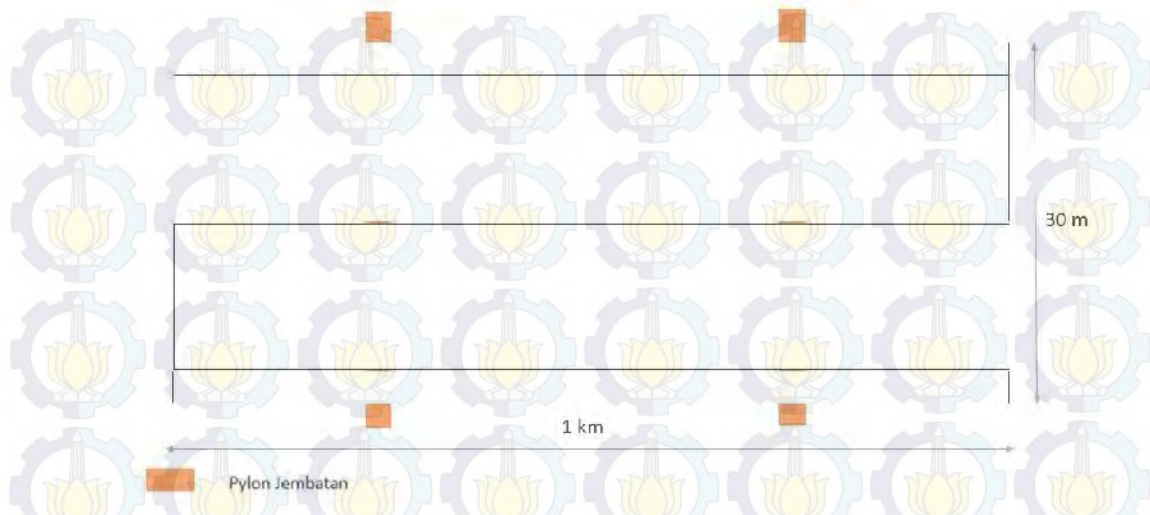
a) SURAMADU tampak atas



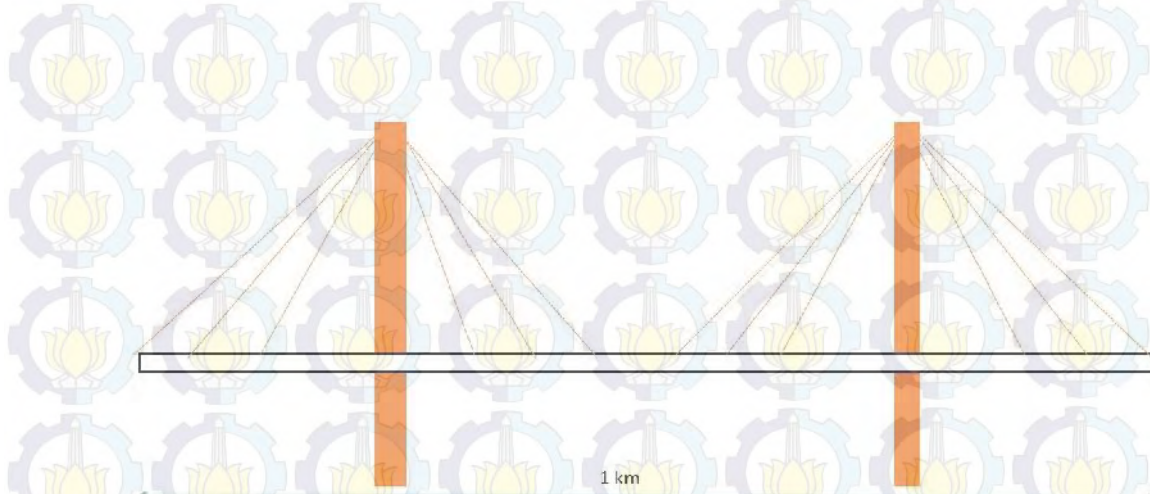
a) SURAMADU tampak samping
Gambar 3.4 Jembatan SURAMADU

3.5.2. Desain Jembatan Dalam Simulasi

Pada simulasi yang akan dilakukan desain jembatan mengacu pada Jembatan Suramadu secara *real*. Jembatan yang akan di simulasikan yakni pada sisi *main bridge*, karena sisi ini merupakan sisi terpenting pada jembatan dan merupakan bagian paling rawan terkena kerusakan. Pada jembatan simulasi diasumsikan panjang jembatan sebesar 1 kilometer dan lebar 30 meter. Berikut pada Gambar 3.5 ditampilkan desain jembatan yang akan digunakan dalam simulasi.



a) Desain jembatan dalam simulasi tampak atas



b) Desain jembatan dalam simulasi tampak samping

Gambar 3.5 Desain jembatan dalam simulasi

3.5.3. Desain peletakan node

Untuk desain peletakan node, node dibagi menjadi dua macam node yakni node koordinator atau *sink* sebagai pengumpul data dari node sensor yang bertransmisi dan node sensor berfungsi sebagai pengolah data pembacaan sensor dan mengirimkan data pada node koordinator. Node koordinator dikondisikan berada di setiap pylon. Selanjutnya untuk asumsi peletakan sensor, terdapat dua macam asumsi. Asumsi pertama jarak antar node berjarak 100 meter dan asumsi kedua jarak antar node 200 meter. Asumsi ini dibuat agar jarak antar node tidak

melebihi jarak maksimum pada NS2 yakni 250 meter. Dan untuk letak pylon 250 dari sisi samping jembatan.

Pada asumsi yang digunakan karakteristiknya didasarkan pada kendaraan yang melewati jembatan. Dimisalkan terdapat mobil yang bergerak dengan kecepatan 80 km/jam maka untuk perhitungan pengiriman paket dirinci sebagai berikut :

- Kecepatan mobil 80 km/jam = 22,22 m/s
- Waktu yang dibutuhkan mobil untuk melewati jembatan = $\frac{1.000}{22,22} = 45$ detik

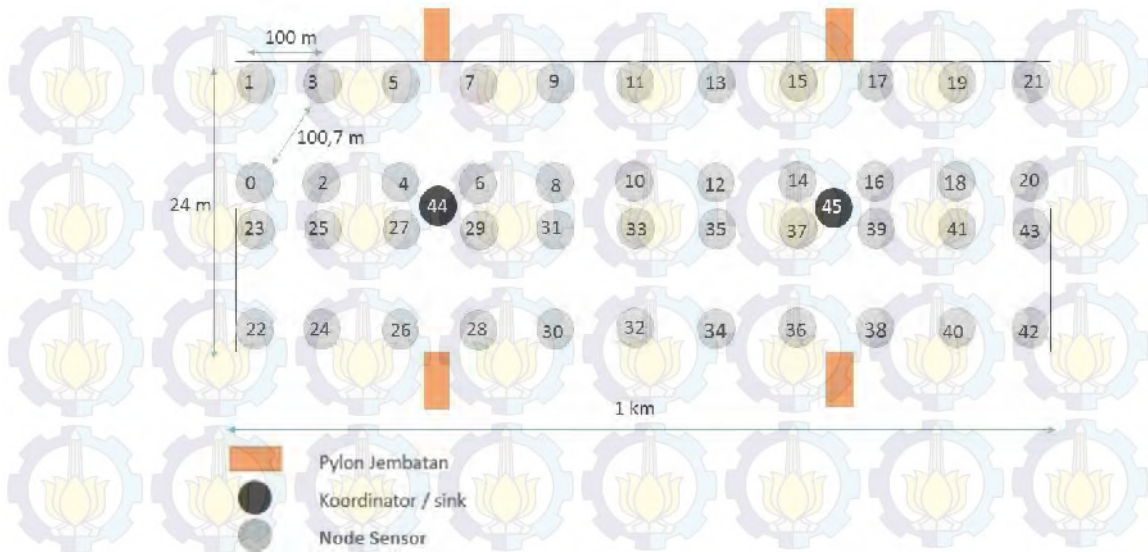
Karena pada desain terdapat dua lajur, yakni lajur dari sisi kiri ke kanan dan lajur dari sisi kanan ke kiri. Pada setiap lajur terdapat 2 node yang diletakkan pada posisi yang bersebrangan, sehingga ketika mobil melaju dan berada pada posisi koordinat yang sama dengan 2 node tersebut maka 2 node tersebut akan mengirimkan data secara bersamaan sehingga dalam 45 detik 2 node akan mengirimkan paket data selama 9 detik. Berikut perhitungannya :

- Waktu pengiriman paket data = $\frac{45}{5} = 9$ detik

Desain asumsi akan diperinci pada penjelasan di bawah ini.

1) Asumsi pertama

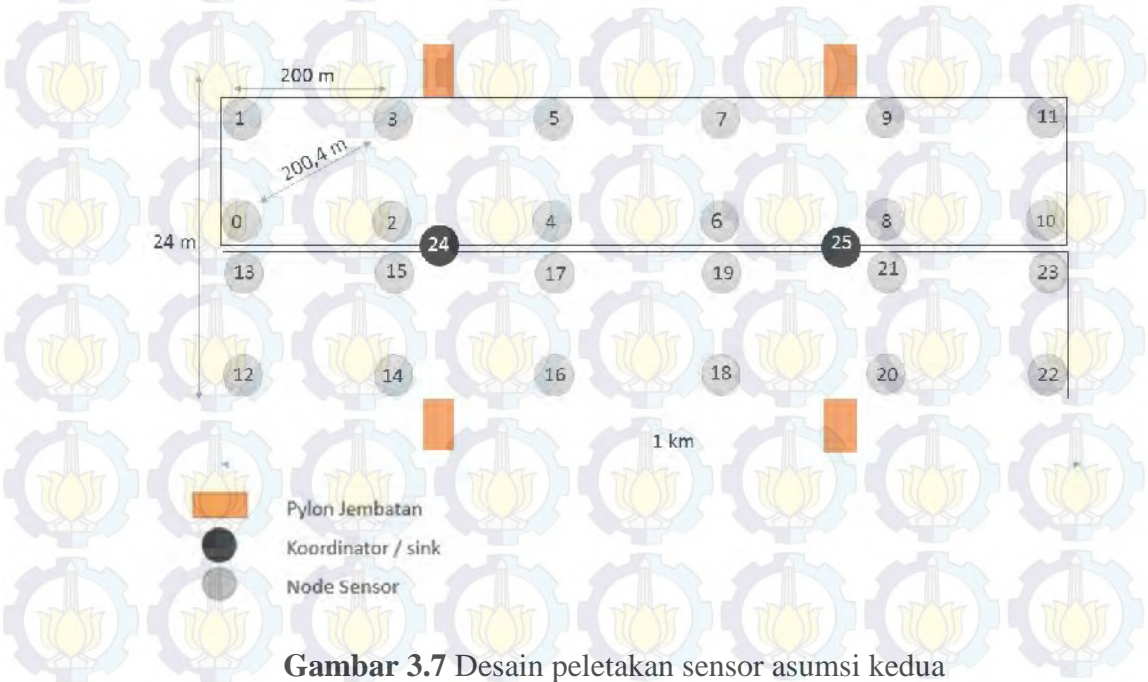
Pada asumsi pertama ini node disebar sebanyak 46 node yang terdiri dari 44 sensor node dan 2 *sink* atau koordinator. Untuk letak node didesain berjarak 100 meter untuk posisi horizontal dan secara vertikal berjarak 12 meter. Sehingga ketika node berkomunikasi secara diagonal jarak antar node sebesar 100,7 meter dan tidak lebih dari jarak maksimumnya.



Gambar 3.6 Desain peletakan sensor asumsi pertama

2) Asumsi kedua

Pada asumsi kedua ini node disebar sebanyak 26 node yang terdiri dari 24 sensor node dan 2 *sink* atau koordinator. Untuk letak node didesain berjarak 200 meter untuk posisi horizontal dan secara vertikal berjarak 12 meter. Sehingga ketika node berkomunikasi secara diagonal jarak antar node sebesar 200,4 meter dan tidak lebih dari jarak maksimumnya. Pada asumsi kedua dapat mengurangi jumlah node yang digunakan, sehingga jumlah node yang digunakan dapat diminimalkan.



Gambar 3.7 Desain peletakan sensor asumsi kedua

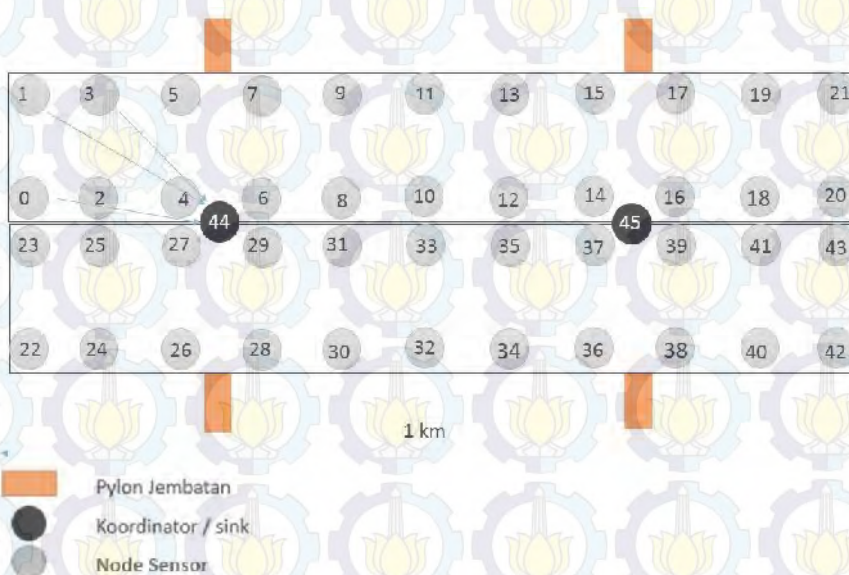
3.6. Desain Topologi

Gambar 3.6 dan Gambar 3.7 menunjukkan desain penyebaran letak sensor pada jembatan dari sudut tampak atas. Dengan mengacu penelitian [17] yakni menggunakan topologi star, mesh dan tree. berdasarkan penelitian [18] [19] sebagaimana telah disebutkan bahwa node di bagi menjadi node sensor dan node koordinator.

Desain topologi yang di tampilkan juga terdapat dua macam berdasarkan desain asumsi yang telah di rancang.

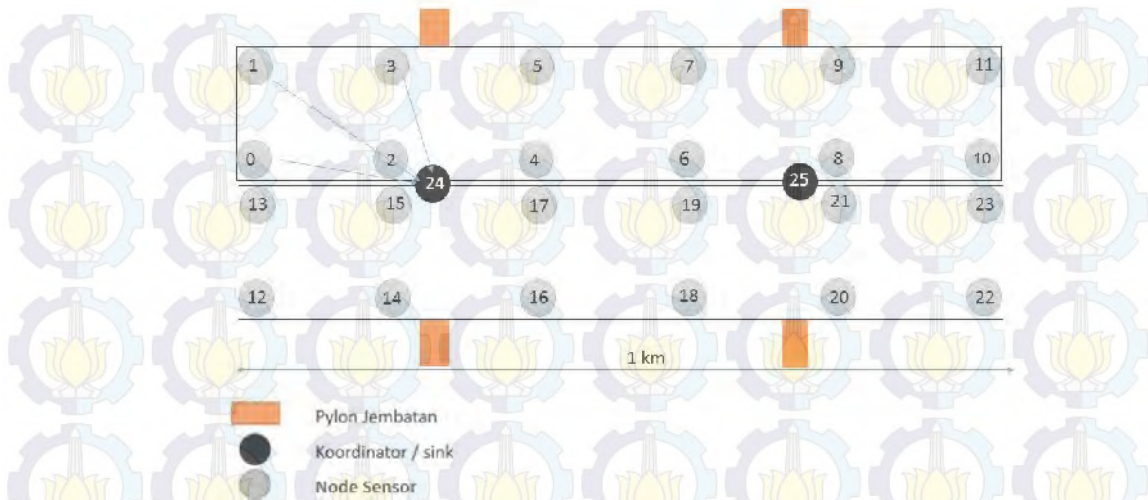
3.6.1. Topologi Star

Topologi Star di desain yakni dengan mengansumsikan node koordinator/*sink* berada di tengah-tengah yakni pada setiap pilon, sedangkan untuk node sensor dapat berkomunikasi secara langsung dengan sink. Desain topologi star dapat dilihat pada Gambar 3.8 dan Gambar 3.9. Berikut pada Gambar 3.8 untuk asumsi pertama dengan jarak antar node sebesar 100 meter.



Gambar 3.8 Desain topologi star asumsi pertama

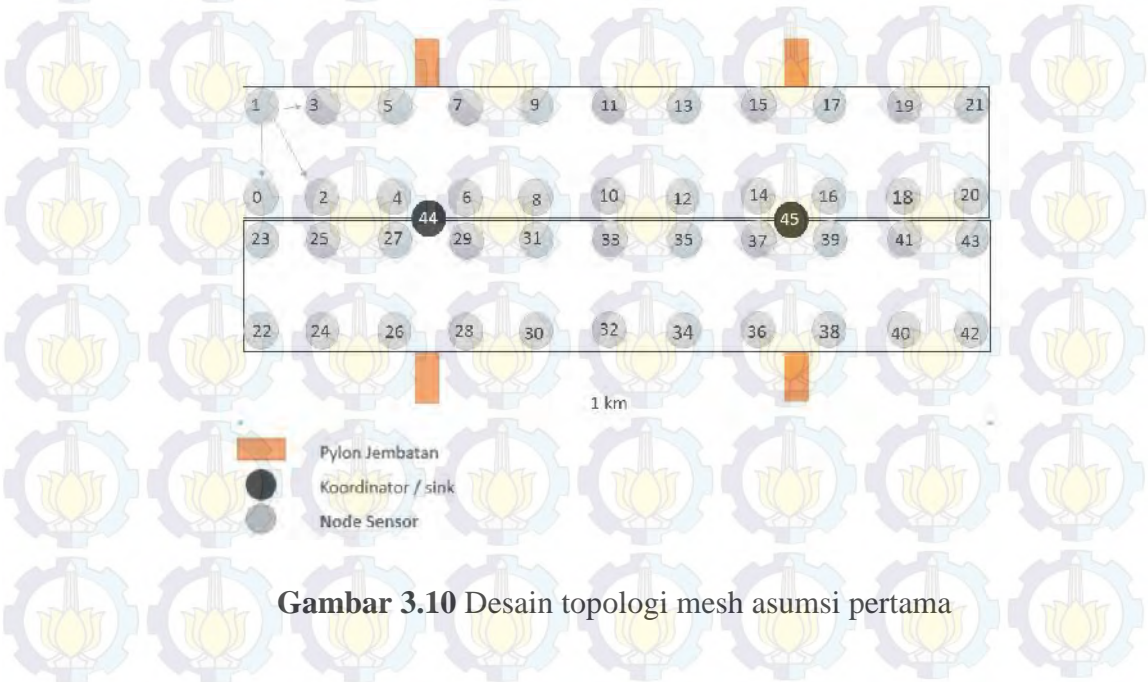
Sedangkan berikut untuk asumsi kedua dengan jarak antar node sebesar 200 meter.



Gambar 3.9 Desain topologi star asumsi kedua

3.6.2. Topologi Mesh

Topologi mesh di desain yakni dengan mengansumsikan node koordinator/*sink* berada di tengah-tengah yakni pada setiap pylon, sedangkan untuk node sensor dapat saling berkomunikasi dimana setiap node sensor memiliki tiga kemungkinan arah komunikasi dengan sensor di sekelilingnya. Desain topologi mesh dapat dilihat pada Gambar 3.10 dan Gambar 3.11. Berikut pada Gambar 3.10 untuk asumsi pertama dengan jarak antar node sebesar 100 meter.



Gambar 3.10 Desain topologi mesh asumsi pertama

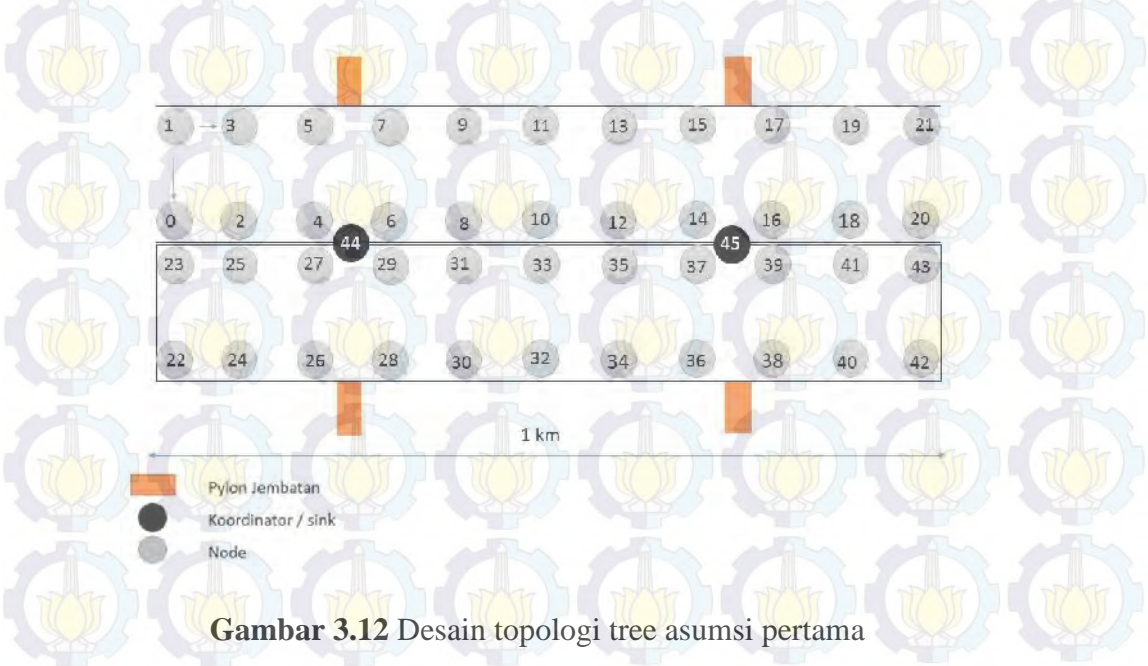
Sedangkan berikut untuk asumsi kedua dengan jarak antar node sebesar 200 meter.



Gambar 3.11 Desain topologi mesh asumsi kedua

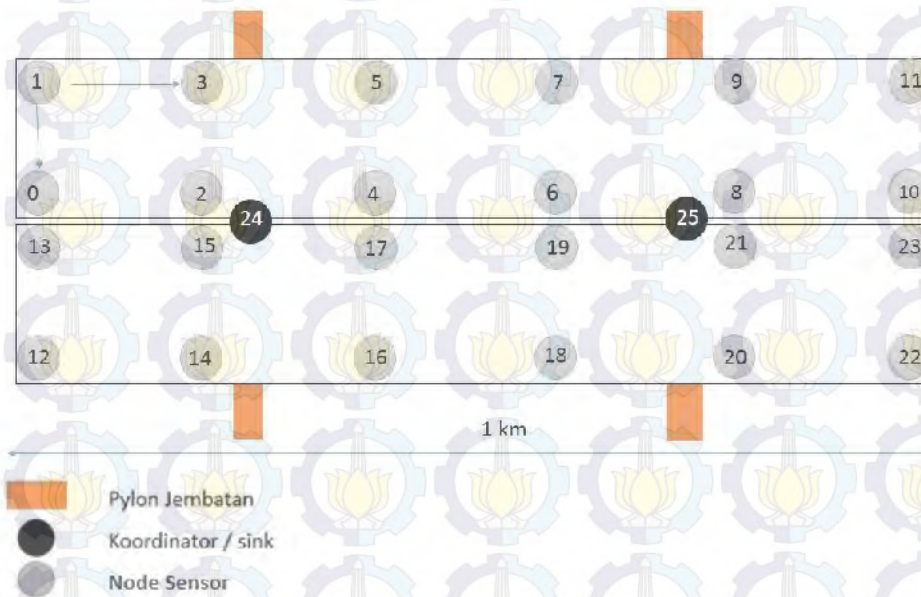
3.6.3. Topologi Tree

Topologi tree di desain yakni dengan mengansumsikan node koordinator/sink berada di tengah-tengah yakni pada setiap pylon, sedangkan untuk node sensor dapat saling berkomunikasi dimana setiap node sensor memiliki dua kemungkinan arah komunikasi dengan sensor di sekelilingnya. Desain topologi mesh dapat dilihat pada pada Gambar 3.12 dan Gambar 3.13. Berikut pada Gambar 3.12 untuk asumsi pertama dengan jarak antar node sebesar 100 meter.



Gambar 3.12 Desain topologi tree asumsi pertama

Sedangkan berikut untuk asumsi kedua dengan jarak antar node sebesar 200 meter.



Gambar 3.13 Desain topologi tree asumsi kedua

3.7. Routing Protokol

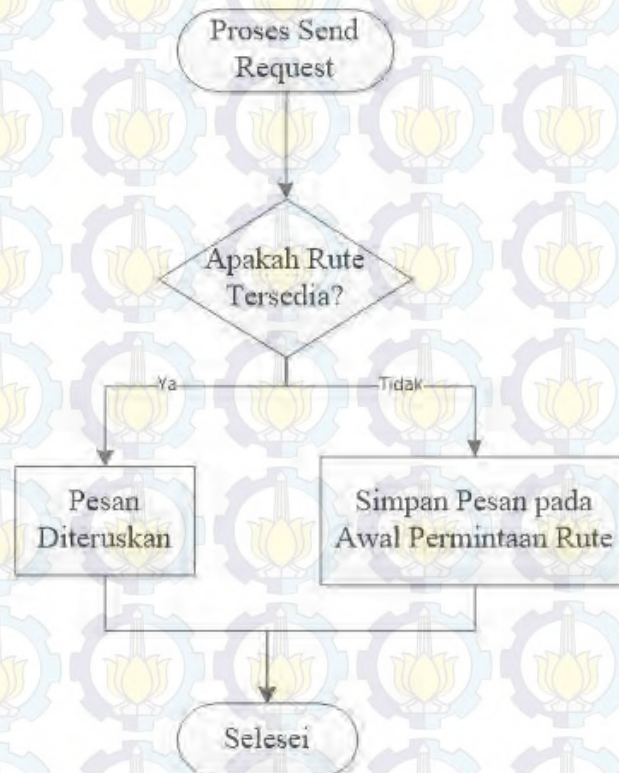
Ada beberapa macam routing protokol yang dipergunakan dalam pengerjaan penelitian ini. Routing protokol tersebut dibagi menjadi dua tipe, yaitu proaktif dan reaktif. Routing protokol bersifat On-demand yang berarti hanya membentuk sebuah rute dari node sumber menuju node tujuan berdasarkan permintaan dari node sumber tersebut. Setiap node menyimpan tabel yang berisi informasi rute ke setiap node yang diketahuinya. Informasi pada setiap node di update jika terjadi perubahan link. Penggunaan protokol routing proaktif secara mendasar memberikan solusi terpendek end-to-end delay, karena informasi routing selalu tersedia dan diperbaharui secara berkala dibandingkan protokol routing reaktif [20].

Routing protokol yang digunakan pada penelitian ini adalah AODV dan DSDV. AODV merupakan salah satu protokol routing reaktif, dimana selama koneksi rute dari pengirim ke penerima telah valid, AODV tidak melakukan pencarian lagi. AODV memelihara rute selama dibutuhkan. Sedangkan DSDV termasuk salah satu protokol proaktif. DSDV juga merupakan salah satu

protokol yang menjaga informasi dalam bentuk tabel (tabel routing) pada setiap node. Tabel tersebut akan sering di update untuk menjaga kekonsistenan dan keakurasian informasi keadaan jaringan. Berikut penjelasan kinerja dari routing protokol AODV dan DSDV.

3.7.1. AODV

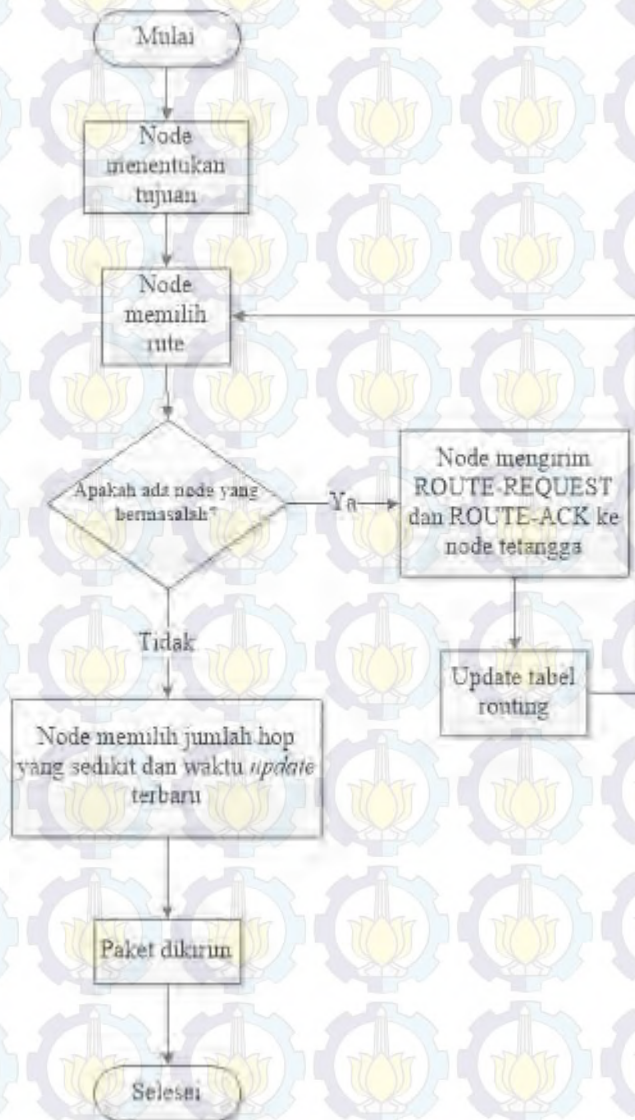
Proses pengiriman paket pada AODV dijelaskan pada Gambar 3.14. Gambar 3.14 menjelaskan proses routing paket pada AODV yang diawali dengan proses permintaan pengiriman update rute. Selanjutnya update rute yang telah dikirim di amati apakah tersedia rute yang bisa dilalui, apabila tidak tersedia maka paket disimpan pada awal permintaan rute namun apabila tersedia rute maka paket diteruskan.



Gambar 3.14 Paket processing AODV [9]

3.7.2. DSDV

Pada DSDV pengiriman paket rendah, faktanya disebabkan karena menggunakan rute link yang rusak [21] [22]. Di DSDV keberadaan stale route atau rute kadaluarsa bukan berarti bahwa tidak ada rute yang valid ke tujuan. Paket dapat diteruskan melalui node tetangga lain yang mungkin memiliki rute ke node tujuan. Pengiriman paket dengan DSDV dapat dilihat pada Gambar 3.15.



Gambar 3.15 Paket *processing* DSDV [9]

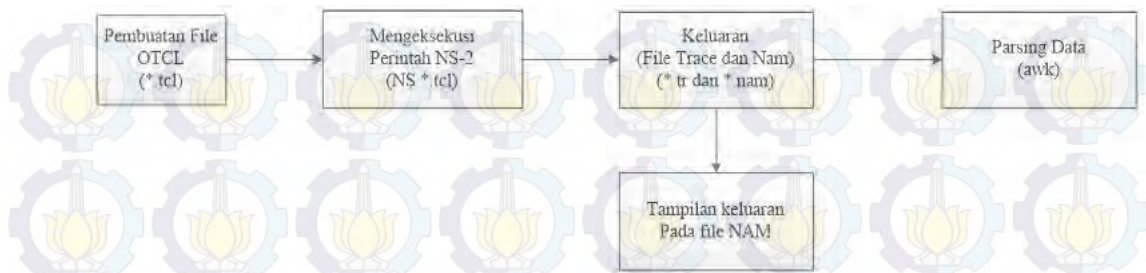
Pada Gambar 3.15 menunjukkan proses kerja dari DSDV. Proses diawali dengan penentuan node sumber dan node tujuan. Selanjutnya node memilih rute yang akan dilewati agar dapat mengirimkan paket sampai ditujuan. Setelah di

dapatkan rute, maka node akan mengecek apakah ada rute yang bermasalah pada suatu node yang akan di lewati. Apabila terdeteksi adanya rute yang bermasalah pada salah satu node yang akan di lewati maka node sumber akan mengirimkan ROUTE-REQUEST dan ROUTE-ACK ke node yang berada disekitar node sumber yang bisasa di sebut node tetangga. Selanjutnya node tetangga akan melakukan update tabel routing yang kemudian dikirimkan kembali ke node sumber. Kemudian node sumber akan melakukan pengecekan kembali terhadap rute yang akan di lewati. Apabila sudah tidak terdeteksi adanya rute yang bermasalah, maka node sumber akan melakukan inisialisasi pada tabel routing yang di kirimkan oleh node tetangga. Node sumber memilih node selanjutnya yang akan di lewati berdasarkan jumlah hop yang sedikit yang waktu update rute terbaru. Setelah di dapatkan node yang pas dengan jumlah hop terkecil, maka paket di kirimkan ke tujuan melalui rute baru yang telah terbentuk

3.8. Model Simulasi

Simulasi dilakukan untuk mengetahui kinerja dari dua routing protokol dengan menggunakan topologi star, mesh dan tree untuk sistem monitoring kesehatan struktur pada jembatan dengan menggunakan NS2. Protokol routing yang digunakan adalah AODV dan DSDV.

Gambar 3.16 menunjukkan keseluruhan proses bagaimana sebuah simulasi jaringan dilakukan dengan NS-2. Keluaran dari NS-2 disebut NAM. NAM menunjukkan pergerakan node dan komunikasi yang terjadi antara node satu dengan node lainnya dalam berbagai kondisi. Dan keluaran lainnya yakni file Trace yang berisi rekaman kejadian yang terjadi yang di simpan pada file *.tr. File keluaran seperti file trace harus diparsing untuk mengekstrak informasi. Parsing dapat dilakukan dengan menggunakan perintah awk. Untuk hasil simulasi di tampilan pada sebuah GUI yang berisi animasi yang di sebut NAM (Network Animator).



Gambar 3.16 Proses simulasi pada NS2 [11]

3.8.1. Format Trace File

Trace file merupakan pencatatan seluruh *event* (kejadian) yang dialami oleh suatu simulasi paket pada simulasi yang dibangun. Pembuatan trace file dilakukan dengan memanggil obyek trace pada library [27]. Gambar 3.17 menunjukkan format isi trace file.

Event	Time	From Node	To Node	Packet Type	Flags	Fid	Source Address	Destination Address	Sequence Number	Packet Id
-------	------	-----------	---------	-------------	-------	-----	----------------	---------------------	-----------------	-----------

Gambar 3.17 Format Trace File

1. Event (kejadian)

Event adalah kejadian yang terjadi dalam simulasi. Kejadian yang dicatat oleh NS yaitu :

r adalah receive yaitu paket yang diterima oleh Node yang dituju

+ adalah enqueue yaitu paket yang masuk dalam antrian atau keluar dari node sumber

- adalah dequeue yaitu paket yang keluar dari antrian

d adalah drop yaitu paket yang di-drop dari antrian

2. Time

Time adalah waktu terjadinya suatu kejadian dalam detik.

3. From Node

From node adalah node sumber.

4. To Node

To node adalah node tujuan.

5. Packet Type

Packet type adalah tipe paket yang dikirim seperti UDP, TCP dan ACK.

6. Packet Size

Packet size adalah ukuran paket dalam byte.

7. Flag

Flag digunakan dalam penanda. Macam-macam flag yang bias digunakan adalah :

E : untuk terjadi kongesti (Congestion Experience/CE)

N : untuk indikasi ECT (ECN-Capable-Transport) pada header IP

C : untuk ECN-Echo

A : untuk pengurangan window kongesti pada header TCP

P : untuk prioritas

F : untuk TCP fast start

8. Fid

Fid adalah penomoran unik dari tiap aliran data

9. Source Address

Source address adalah alamat asal paket.

10. Destination Address

Destination address adalah alamat tujuan paket.

11. Sequence Number

Sequence number adalah nomor urut tiap paket

12. Packet Id

Packet id adalah penomoran unik tiap paket

Berikut contoh hasil trace file dari salah satu asumsi yang dilakukan. Trace file ini nantinya akan di-*parsing* sehingga dapat dianalisa parameter unjuk kinerja dari jaringan yang digunakan.

```
s 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24 0:0 32 0] [0] 0 0  
r 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24 0:0 32 0] [0] 0 0  
s 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24 0:0 32 3] [0] 0 0
```

Gambar 3.18 Contoh trace file pada topologi mesh asumsi pertama

Pada Gambar 3.18, apabila ditulis ulang maka akan nampak seperti berikut

```
s 0.000000000 _23_ AGT --- 0 cbr 100 [0 0 0 0] [energy  
19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] ----- [23:24  
0:0 32 0] [0] 0 0
```

keterangan :

- s merupakan event / kejadian mengirimkan paket data.
- 0.000000000 merupakan waktu kejadian pengiriman paket data yaitu pada 0.000000000 detik.
- _23_ merupakan node sumber dimana dalam hal ini node 23 merupakan node pengirim paket data.
- AGT merupakan application layer dimana dalam hal ini pada node 23 terjadi pengiriman paket data dari application layer.
- --- menyatakan tidak adanya flag sebagai penanda.
- 0 merupakan penomoran unik dari aliran data.
- CBR merupakan tipe paket yang dikirim oleh application layer node 23.
- 100 merupakan ukuran paket CBR dalam byte yang dikirimkan oleh application layer node 23.
- [0 0 0 0] merupakan penanda bahwa belum terjadi routing.

- [energy 19906.000000 ei 0.000 es 0.000 et 0.000 er 0.000] menandakan bahwa energi dari node 23 pada saat pengiriman paket data CBR sebesar 100 byte adalah 19906 Joule dengan energi idle yang memiliki nilai 0.000 Joule, energi sleep 0.000 Joule, energi transmit 0.000 Joule, dan energi receive 0.000.
- ----- menandakan tidak terjadi flag.
- [23:24 0:0 32 0] dimana 13:24 menandakan node 23 mengirimkan paket data pada port 0 menuju node 24 dengan port 0, dengan TTL 32 hop, dan 0 menandakan belum ada hop selanjutnya.
- [0] menandakan penomoran unik dari aliran data.
- 0 0 menandakan tahap proses pengiriman node.

3.9. Parameter Kinerja Simulasi

Beberapa parameter yang akan diukur sebagai analisa kinerja sistem yakni :

1. Energi

Energi merupakan kemampuan node saat proses komunikasi data. Energi yang dihasilkan adalah keseluruhan energi yang digunakan baik saat mengirim data ataupun saat membroadcast data dari node tetangganya.

$$\text{Energi} = E_0 - E_t \quad (3.4)$$

Keterangan :

Energi = Energi (Joule)

E_0 = Energi awal sebelum pengiriman paket (Joule)

E_t = Energi akhir setelah pengiriman paket (Joule)

2. Packet Loss

Packet Loss adalah parameter yang menggambarkan suatu kondisi yang menunjukkan jumlah total paket yang hilang. Kegagalan tersebut dapat disebabkan oleh beberapa kemungkinan, diantaranya yaitu:

- a. Terjadinya *overload* trafik didalam jaringan,
- b. Tabrakan dalam jaringan,
- c. *Error* yang terjadi pada media fisik,
- d. Kegagalan yang terjadi pada sisi penerima antara lain dapat disebabkan karena *overflow* yang terjadi pada *buffer*.

Packet loss dapat dirumuskan seperti pada persamaan :

$$PL = Pl_s - Pl_r \quad (3.5)$$

Keterangan :

Pl = Banyak paket loss (paket)

Pl_s = Banyak paket yang dikirim (paket)

Pl_r = Banyak paket yang diterima (paket)

3. Throughput

Throughput merupakan jumlah total kedatangan paket yang sukses yang diamati pada tujuan selama interval waktu tertentu dibagi oleh durasi interval waktu tersebut. Throughput juga dapat di definisikan sebagai kecepatan transfer data yang diukur dalam satuan bit per sekon (bps). Throughput dapat dirumuskan seperti pada persamaan:

$$\text{Throughput} = \frac{Pl_r}{t} \quad (3.6)$$

Keterangan :

Throughput = Throughput (bps)

Pl_r = Banyak paket yang diterima (paket)

t = Waktu pengambilan sampel (detik)

4. Delay

Delay atau waktu tunda adalah interval waktu yang dibutuhkan paket data untuk menempuh jarak dari data mulai di kirim sampai dengan data sampai

ditujuan. *Delay* dapat dipengaruhi oleh jarak, media fisik, atau juga waktu proses yang lama. *Delay* dinyatakan dalam satuan detik atau *second*. Perhitungan *delay* di dapat dari mengurangkan waktu saat pengiriman paket data dengan waktu saat paket data di terima. *Delay* dapat dirumuskan seperti pada persamaan:

$$\text{Delay} = t_s - t_r \quad (3.7)$$

Keterangan :

Delay = Delay (s)

t_s = Waktu pengiriman paket data (s)

t_r = Waktu penerimaan paketdata (s)

3.10. Parameter Simulasi

Parameter yang digunakan dalam simulasi adalah parameter dengan menggunakan data sheet dari XBeePro series 1 dan disesuaikan dengan software NS-2. Parameter simulasi tersebut dapat dilihat pada Tabel 3.1

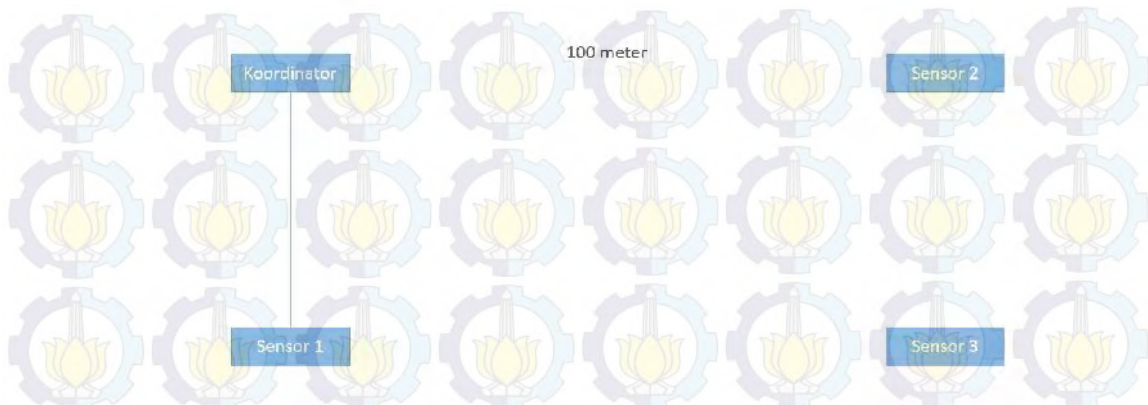
Tabel 3.1 Parameter simulasi

Parameter	Nilai
Jenis Kanal	Wireless Channel
Jenis Layer Fisik dan MAC	802.15.4
Jenis Baterai yang digunakan	Imedion 9.6 V, 220 mA

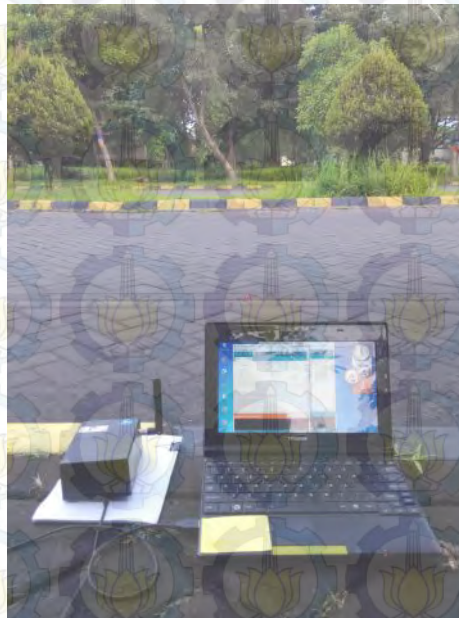
3.11. Validasi Parameter Simulasi

Validasi ini dilakukan untuk memvalidasikan parameter-parameter yang telah diasumsikan dengan mengambil beberapa titik node saja.

Berikut desain untuk validasi parameter simulasi secara *real time*. Pada saat validasi dilakukan ada tiga node sensor dan satu koordinator. Dimana jarak sebesar 12 meter dan 100 meter.



Gambar 3.19 Desain implementasi sistem untuk validasi parameter simulasi dengan jarak 100 meter



Gambar 3.20 Implementasi sistem untuk validasi parameter simulasi

3.11.1. Perangkat Validasi

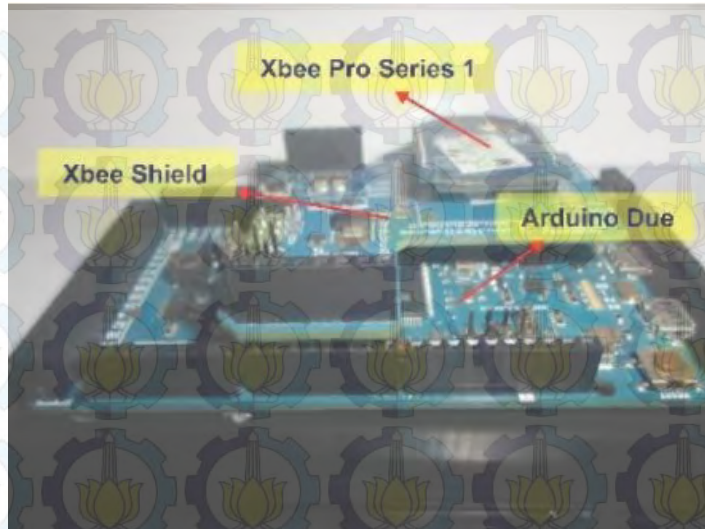
Dalam proses validasi parameter simulasi atau implementasi dari hasil simulasi menggunakan perangkat keras yang terdiri dari dua bagian, yaitu bagian sensing atau node sensor sebagai pengambil data sensor dan bagian pengolahan data atau node koordinator/*sink*. Setiap node menggunakan mikrokontroler arduino duemilanove sebagai pengolah data dan modul komunikasi *Xbee* untuk pentransmisiannya. Sedangkan untuk proses sensing menggunakan sensor accelerometer MMA 7361.



Gambar 3.21 Bentuk fisik node

➤ **Node Koordinator**

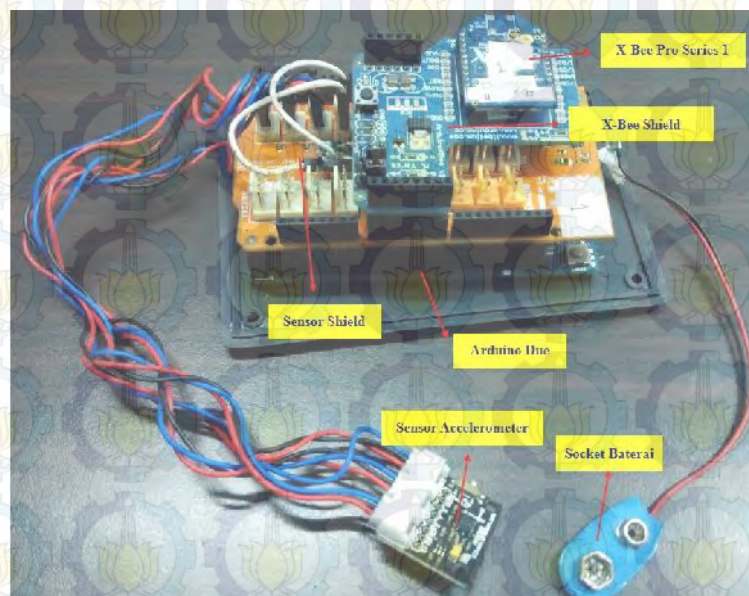
Node koordinator sebagai penerima data dan pengolah data dari node sensor serta yang selanjutnya akan dikirimkan ke *base station*. Node koordinator terdiri dari mikrokontroler arduino due sebagai pengolah data serta *Xbee* sebagai modul komunikasi untuk dapat berkomunikasi dengan perangkat pada sisi node sensor. Pada sisi koordinator tidak menggunakan ataupun terhubung dengan sensor karena pada sisi koordinator ini hanya berfungsi sebagai penerima data dari node sensor. Pada node koordinator menggunakan mikrokontroler arduino duemilanove sebagai pengolah data dan modul komunikasi xbee untuk dapat terhubung secara nirkabel dengan node sensor. Selanjutnya, modul *Xbee* sebagai modul komunikasi data. *Xbee* yang digunakan adalah *Xbee Pro Series 1*. Modul ini adalah perangkat dengan protokol standart IEEE 802.15.4. *Xbee Pro Series 1* memiliki kemampuan berkomunikasi secara *point-to-point*, dan *point-to-multipoint*. Pada bagian node sensor modul *Xbee* diatur agar dapat melakukan komunikasi data secara *multipoint-to-point*.



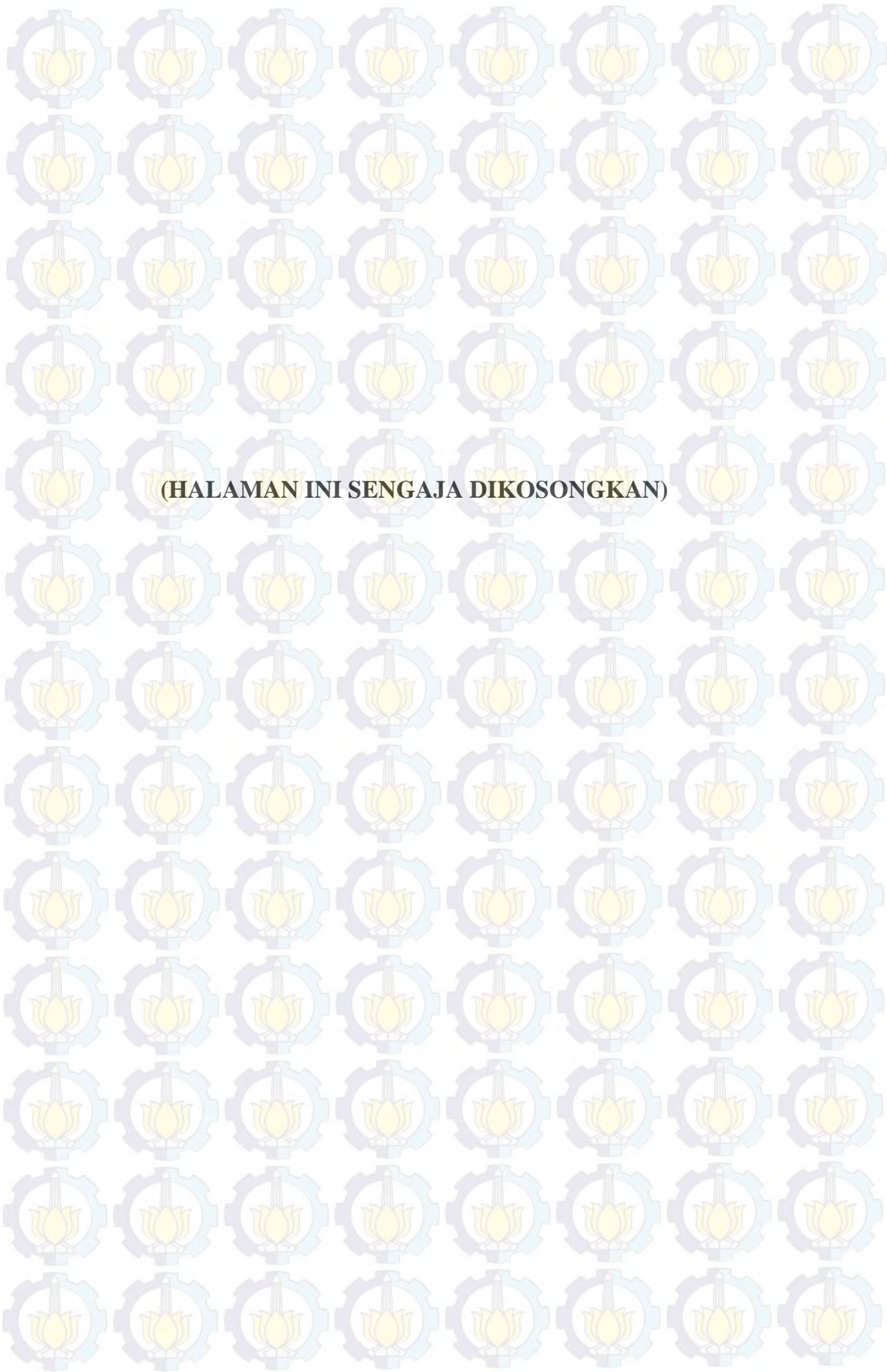
Gambar 3.22 Susunan node koordinator.

➤ **Node Sensor**

Node sensor terdiri dari sensor accelerometer sebagai alat sensing, mikrokontroler arduino due sebagai pengolah data serta Xbee sebagai modul komunikasi untuk dapat terhubung dengan perangkat pada sisi node koordinator.



Gambar 3.23 Susunan node sensor.



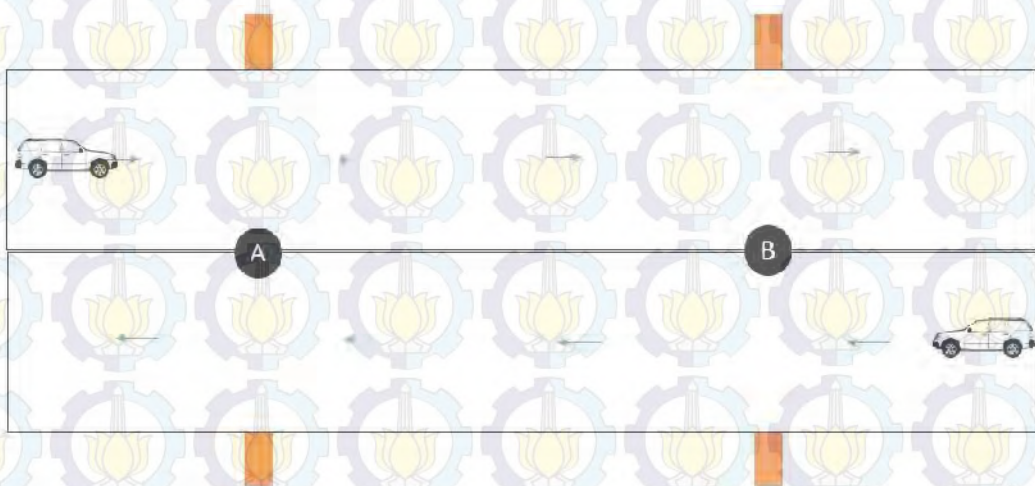
BAB 4 HASIL DAN ANALISA

Pada bab ini membahas mengenai pelaksanaan pengujian yang dilakukan dengan menggunakan skenario yang telah dijelaskan pada bab sebelumnya. Pengujian diawali dengan skenario pengujian terhadap asumsi yang ditetapkan.

4.1. Skenario Pengujian Sistem

Dalam pengujian sistem disini ditekankan pada dua jalur, yakni pada jalur kendaraan roda empat atau lebih baik sisi arah ke madura maupun sisi arah ke surabaya. Skenario yang dirancang menggunakan dua macam skenario yakni skenario dengan asumsi pertama menggunakan jarak 100 meter x 12 meter untuk setiap jalur dan skenario dengan asumsi kedua menggunakan jarak 200 meter x 12 meter untuk setiap jalur. Setiap asumsi juga membandingkan penggunaan routing dan topologi. Untuk routing menggunakan dua macam routing, yakni routing AODV dan DSDV. Sedangkan untuk topologi menggunakan topologi star, tree dan mesh.

Pada skenario pengujian sistem, jembatan dibagi menjadi dua jalur, dimana satu sisi untuk jalur perjalanan kendaraan dari kiri ke kanan dan sisi lainnya untuk jalur perjalanan kendaraan kanan ke kiri.



Gambar 4.1. Pembagian arah jalur pada jembatan

Pembagian rute dan penentuan node koordinator yakni untuk lajur kiri ke kanan menggunakan node koordinator B, sedangkan untuk lajur kanan ke kiri menggunakan node koordinator A.

Pada setiap asumsi yang akan digunakan karakteristiknya didasarkan pada kendaraan yang melewati jembatan. Dimisalkan terdapat mobil yang bergerak dengan kecepatan 80 km/jam maka untuk perhitungan pengiriman paket dirinci sebagai berikut :

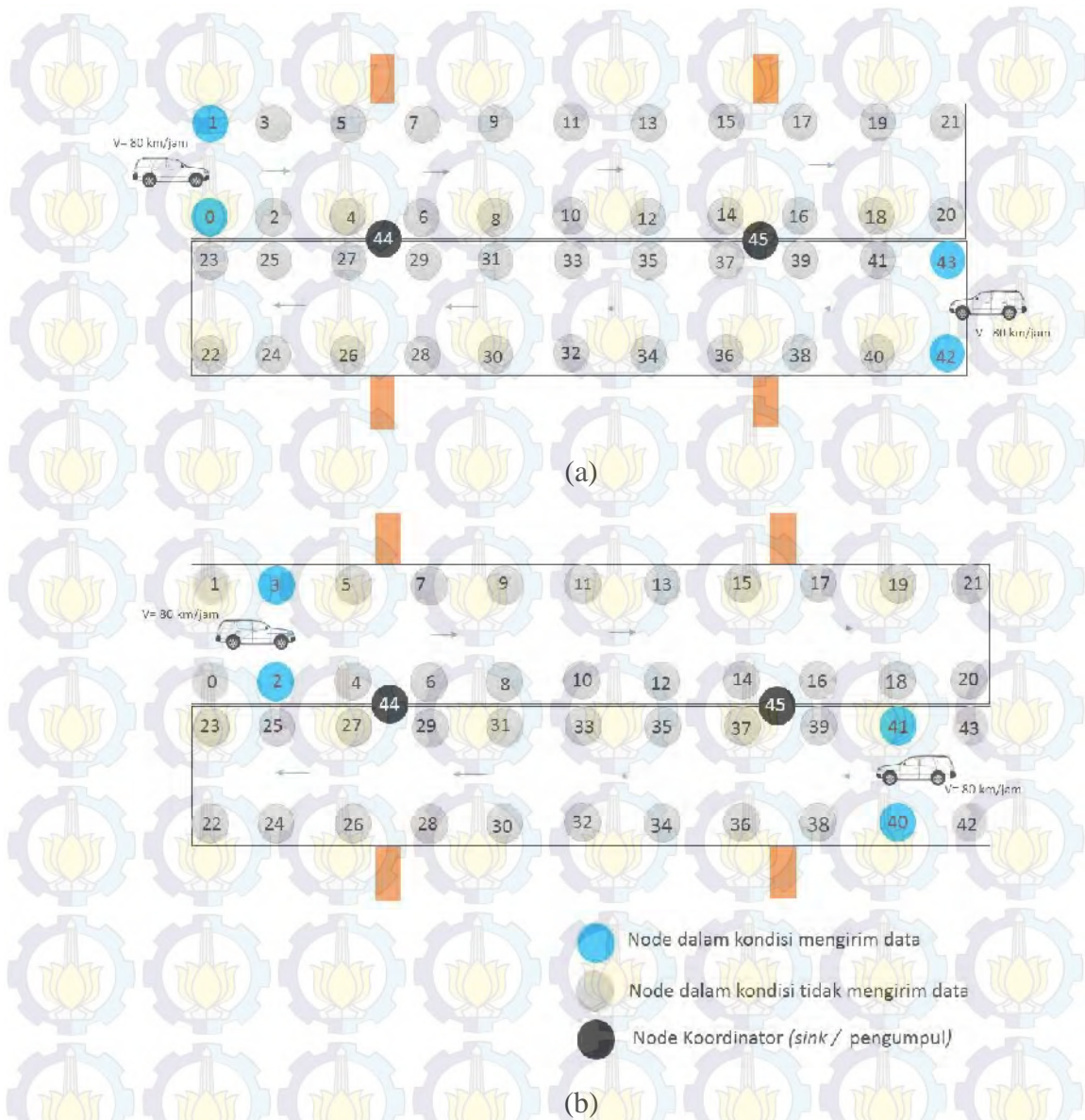
- Kecepatan mobil 80 km/jam = 22,22 m/s
- Waktu yang dibutuhkan mobil untuk melewati jembatan = $\frac{1.000}{22,22} = 45$ detik

Karena pada desain terdapat dua lajur, yakni lajur dari sisi kiri ke kanan dan lajur dari sisi kanan ke kiri. Pada setiap lajur terdapat 2 node yang diletakkan pada posisi yang bersebrangan, sehingga ketika mobil melaju dan berada pada posisi koordinat yang sama dengan 2 node tersebut maka 2 node tersebut akan mengirimkan data secara bersamaan sehingga dalam 45 detik 2 node akan mengirimkan paket data selama 9 detik. Berikut perhitungannya :

- Waktu pengiriman paket data = $\frac{45}{5} = 9$ detik

4.2. Asumsi Pertama

Pada asumsi pertama ini, posisi node di kondisikan jarak antar node terletak dengan jarak 100 meter dan 12 meter. Pada asumsi ini di skenarioikan terdapat sebuah kendaraan yang melintas dengan kecepatan 80 km/jam, sehingga node-node sensor akan mengikuti pergerakan kendaraan dan node yang akan dilewati akan bersiap-siap untuk mengirimkan paket data menuju node koordinator.



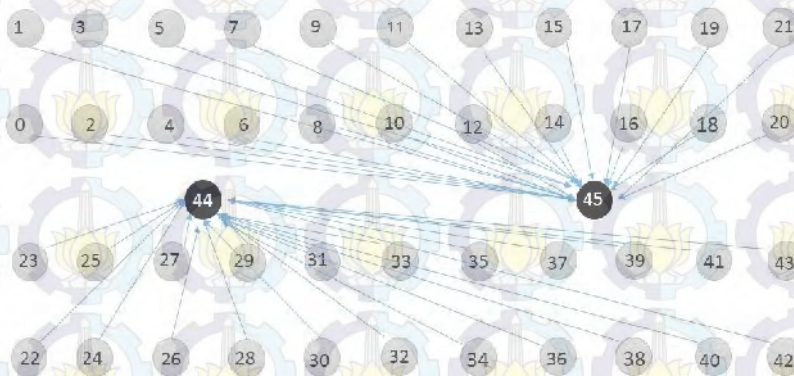
Gambar 4.2. Skenario pada asumsi pertama

Terlihat dari Gambar 4.2 dimana untuk skenario pada asumsi pertama. Pada sisi lajur kiri ke kanan ketika ada kenjaraan melaju memasuki jembatan sehingga posisi koordinatnya kendaraan sama dengan node 0 dan node 1 maka node 0 dan node 1 akan mengirim paket data menuju node 45 (node koordinator). Sedangkan Pada sisi lajur kanan ke kiri ketika ada kenjaraan melaju memasuki jembatan maka node 42 dan node 43 akan mengirim paket data menuju node 44 (node koordinator) seperti pada Gambar 4.2 (a). Node aktif mengirimkan data ditandai dengan node berwarna biru. Pada sisi lajur kiri ke kanan ketika posisi kendaraan

sama dengan node 2 dan node 3 maka node 2 dan node 2 akan mengirim paket data menuju node 45, untuk node 0 dan node 1 sudah tidak lagi mengirimkan paket data. Sedangkan Pada sisi lajur kanan ke kiri ketika posisi kendaraan sama dengan node 40 dan node 41 maka node 40 dan node 41 akan mengirim paket data menuju node 44, untuk node 42 dan node 43 sudah tidak lagi mengirimkan paket data seperti pada Gambar 4.2 (b).

4.2.1. Topologi Star

Topologi star merupakan topologi paling sederhana dimana komunikasi terjadi secara *point-to-point* node dengan sink. Desain topologi seperti Gambar 4.3.



Gambar 4.3. Topologi star asumsi pertama

Terlihat pada Gambar 4.3, bahwa pada topologi star setiap node berkomunikasi secara langsung dengan node koordinator (node 44 dan node 45).

Topologi pada asumsi yang ini mirip dengan Gambar 4.2 dimana setiap node mengirim paket data langsung menuju koordinator. Pada isi kiri ke kanan awalnya, node 0 dan node 1 akan mengirimkan paket data secara bersamaan dan langsung menuju node 45 selama 9 detik, kemudian node 2 dan node 3 akan mengirimkan paket data secara bersamaan dan langsung menuju node 45 selama 9 detik. Dilanjutkan node 4 dan node 5 mengirimkan paket data, lalu node 6 dan node 7, dan yang terakhir adalah node 20 dan node 21 mengirimkan paket data secara bersamaan langsung menuju node 45. Sedangkan Pada sisi kanan kekiri awalnya, node 42 dan node 43 akan mengirimkan paket data secara bersamaan

dan langsung menuju node 44 selama 9 detik, kemudian node 40 dan node 41 akan mengirimkan paket data secara bersamaan dan langsung menuju node 45 selama 9 detik. Dilanjutkan node 38 dan node 39 mengirimkan paket data, lalu node 36 dan node 37, dan yang terakhir adalah node 22 dan node 23 mengirimkan paket data secara bersamaan langsung menuju node 44. Hasil dari simulasi tampak pada Tabel 4.1. dan Tabel 4.2. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

Tabel 4.1. Hasil simulasi topologi star dengan routing DSDV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1.	<i>Throughput</i>	0.65 kbps
2.	<i>Delay end-to-end</i> rata-rata	0.840 ms
3.	<i>Packet Loss</i>	139 paket
4.	Energi <i>end-to-end</i> rata-rata	0.106 Joule

Tabel 4.2. Hasil simulasi topologi *star* dengan routing AODV dengan asumsi pertama

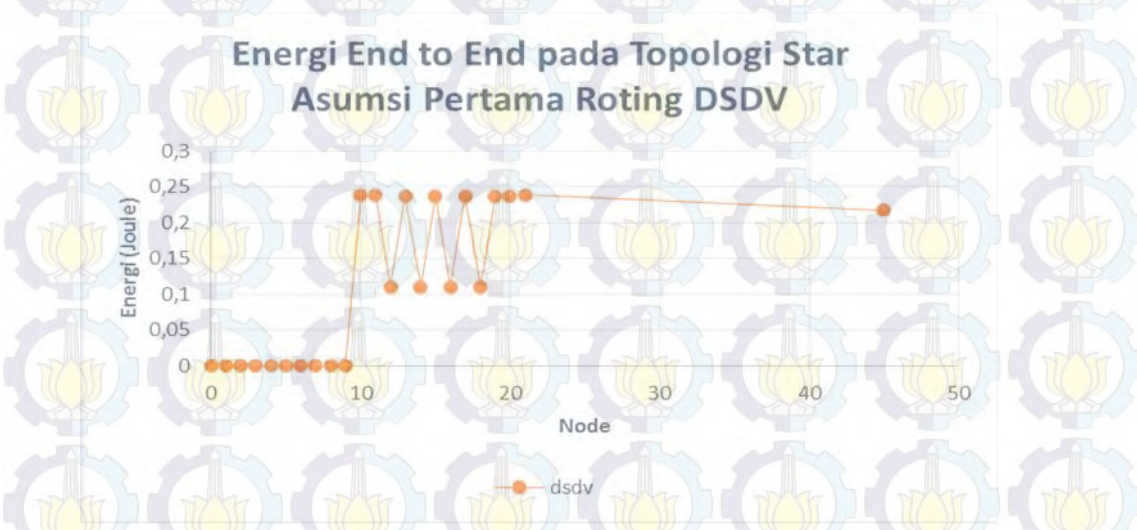
No.	Parameter Kinerja Jaringan	Nilai
1)	<i>Throughput</i>	0.69 kbps
2)	<i>Delay end-to-end</i> rata-rata	0.532 ms
3)	<i>Packet Loss</i>	142 paket
4)	Energi <i>end-to-end</i> rata-rata	0.154 Joule

karena terdapat node yang melebihi jarak jangkauan maka terdapat beberapa paket yang tidak dikirimkan.

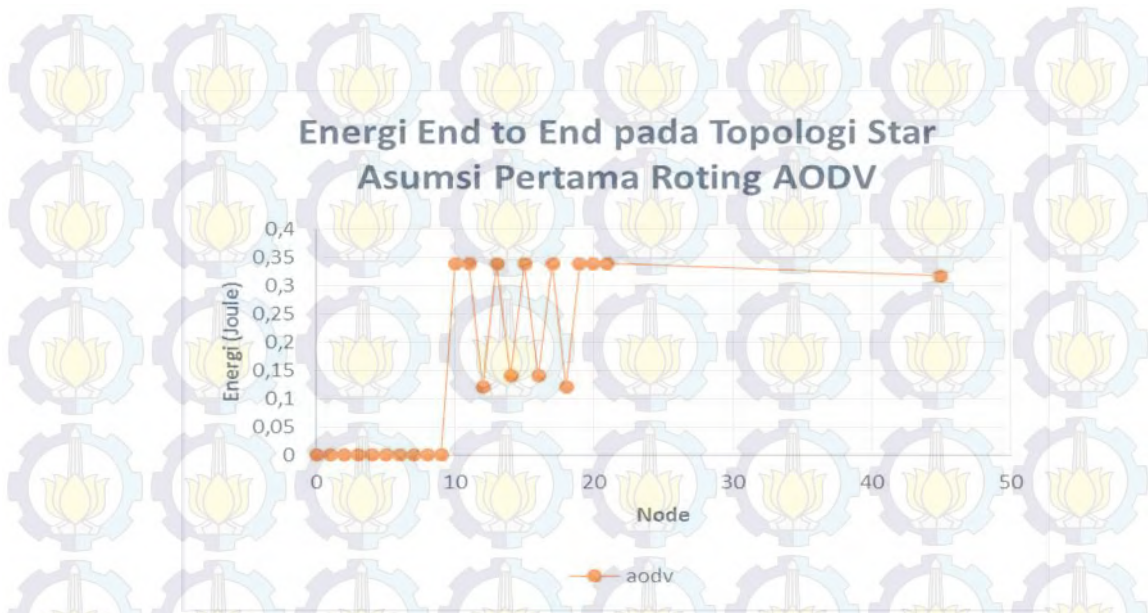
Parameter energi *end-to-end* pada topologi star asumsi pertama untuk routing AODV dan DSDV pada masing-masing node ditunjukkan pada Gambar 4.4 dan Gambar 4.5. Pada lajur kanan ke kiri node 0, node 1, node 2,

node 3, node 4, node 5, node 6, node 7, dan node 8, energi *end-to-end* masing-masing node tersebut adalah nol. Hal ini disebabkan ketika node-node tersebut mengirimkan paket data secara langsung menuju node 0, karena jarak yang melebihi jangkauan XBee maka tidak paket tersebut tidak dikirimkan. Sedangkan pada node 10, node 11, node 12, node 13, node 15, node 17, node 19, node 21, dan node 20, masing-masing memiliki nilai energi *end-to-end* untuk routing AODV berturut-turut 0,338 Joule, 0,339 Joule, 0,338 Joule, 0,338 Joule, 0,338 Joule, 0,338 Joule, 0,339 Joule dan 0,338 Joule, Node – node ini untuk aktifitas “mendengar”. Node 12 dan Node 18 memiliki nilai energi *end-to-end* 0,12. Node 14 dan Node 16 memiliki nilai energi *end-to-end* 0,14 Joule untuk aktifitas pengiriman paket data menuju node 45 (Koordinator). Sedangkan node 45 memiliki nilai energi *end-to-end* 0,317 Joule untuk aktifitas menerima paket data dan “mendengar. Sedangkan untuk routing DSDV berturut-turut 0,238 Joule, 0,239 Joule, 0,238 Joule, 0,238 Joule, 0,238 Joule, 0,238 Joule, 0,239 Joule dan 0,238 Joule. Node 12, Node 14, Node 16 dan Node 18 memiliki nilai energi *end-to-end* 0,11 untuk aktifitas pengiriman paket data menuju node 45 (Koordinator). Sedangkan node 45 memiliki nilai energi *end-to-end* 0,217 Joule untuk aktifitas menerima paket data dan “mendengar.

Berdasarkan Gambar 4.4 dan Gambar 4.5 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 0,154 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 0,106 Joule.



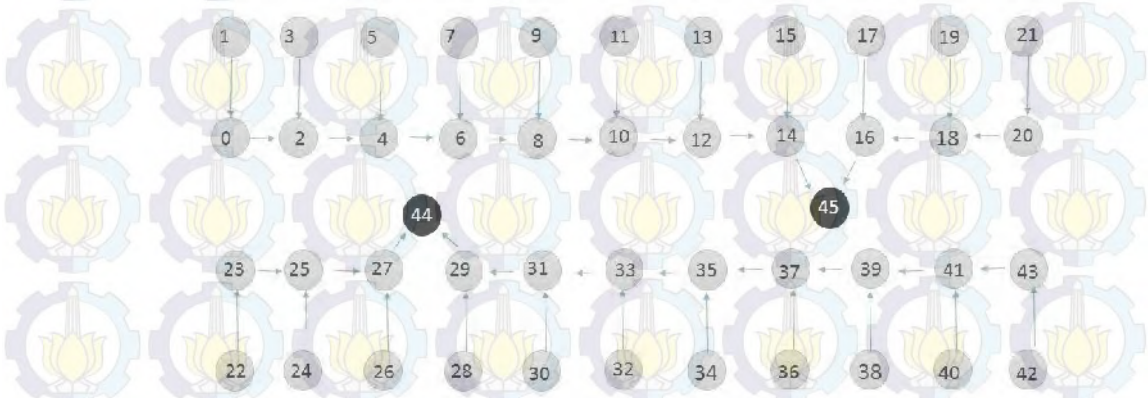
Gambar 4.4. Energi End-to-end topologi star asumsi pertama routing DSDV



Gambar 4.5. Energi End-to-end topologi star asumsi pertama routing AODV

4.2.2. Topologi Mesh

Topologi Mesh atau biasa disebut komunikasi *peer-to-peer* adalah komunikasi yang terjadi antar node Desain topologi seperti Gambar 4.6.



Gambar 4.6. Topologi mesh asumsi pertama

Terlihat pada Gambar 4.6 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya.

Pada skenario dengan asumsi pertama ini, pada sisi kiri ke kanan, ketika sebuah mobil memiliki koordinat yang sama dengan node 1 dan node 0, maka node 1 mengirimkan data melalui node 0 yang kemudian di teruskan dari node 0

menuju node 2. Node 2 juga menerima data dari node 3 dan seterusnya sehingga data sampai di node 45. Sedangkan pada sisi kanan ke kiri, ketika sebuah mobil memiliki koordinat yang sama dengan node 42 dan node 43, maka node 42 mengirimkan data melalui node 43 yang kemudian diteruskan dari node 43 menuju node 41. Node 41 juga menerima data dari node 40 dan seterusnya sehingga data sampai di node 44.

Hasil dari simulasi tampak pada Tabel 4.3. dan Tabel 4.4. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

Tabel 4.3. Hasil simulasi topologi mesh dengan routing DSDV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	7.11 kbps
2	<i>Delay end-to-end</i> rata-rata	27,25 ms
3	<i>Packet Loss</i>	2 paket
4	Energi <i>end-to-end</i> rata-rata	5,09 Joule

Tabel 4.4. Hasil simulasi topologi mesh dengan routing AODV dengan asumsi pertama

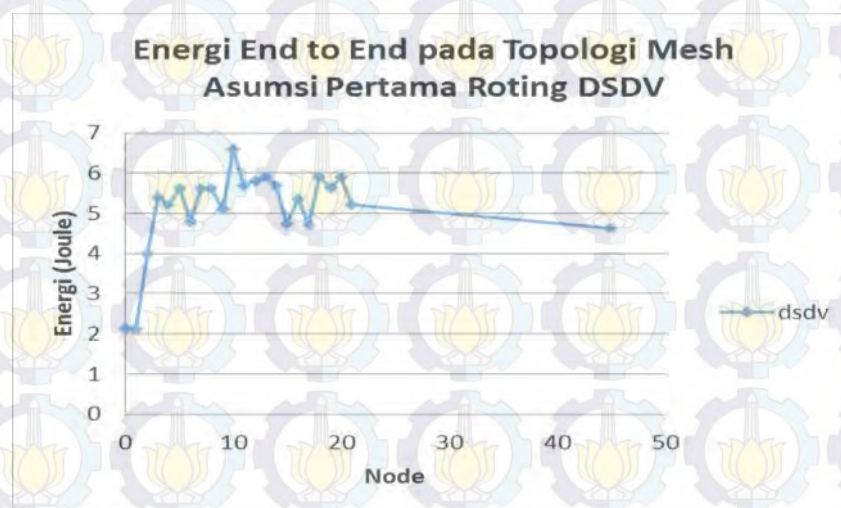
No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	21.28 kbps
2	<i>Delay end-to-end</i> rata-rata	10.68 ms
3	<i>Packet Loss</i>	1 paket
4	Energi <i>end-to-end</i> rata-rata	5.66 Joule

Terlihat pada Gambar 4.7, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,14 Joule, node 1 bernilai 2.11 Joule, node 2 bernilai 4 Joule, node 3 bernilai 5.4 Joule, node 4 bernilai 5.2 Joule, node 5 bernilai 5.6 Joule, node 6 bernilai 4.8 Joule, node 7

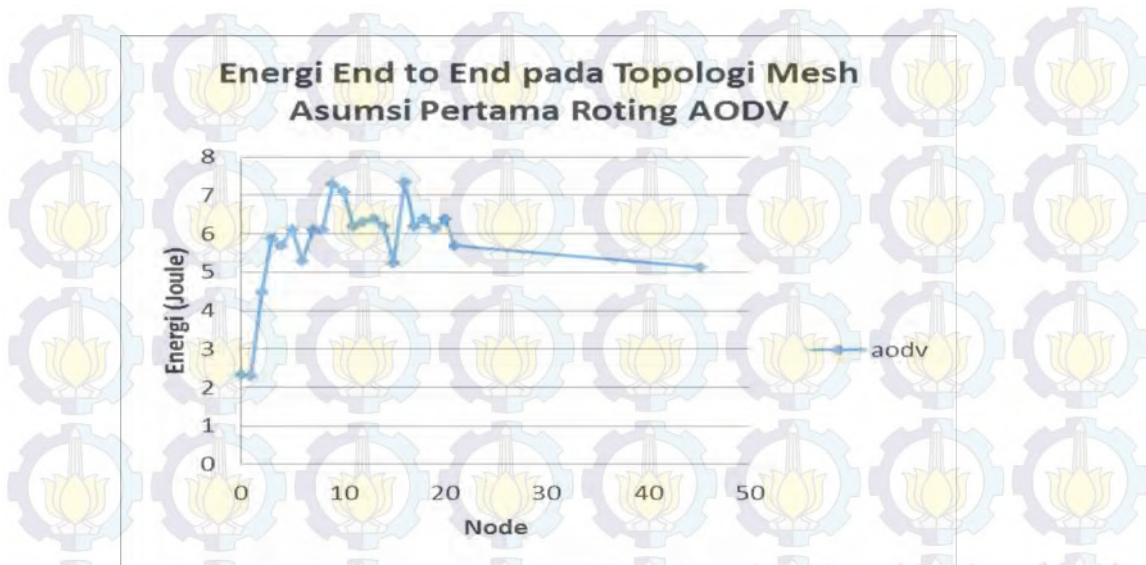
bernilai 5.6 Joule, node 8 bernilai 5.6 Joule, node 9 bernilai 5.1 Joule, node 10 bernilai 6.6 Joule., node 11 bernilai 5.7 Joule, node 12 bernilai 5.8 Joule, node 13 bernilai 5.9 Joule, node 14 bernilai 4.74 Joule, node 15 bernilai 5,34 Joule dan node 16 bernilai 4.7 Joule, node 17 bernilai 5.9 Joule, node 18 bernilai 5.63 Joule, node 19 bernilai 5.9 Joule, node 20 bernilai 5.4 Joule, node 21 bernilai 5.2 Joule dan node 45 bernilai 4.63 Joule.

Terlihat pada Gambar 4.8, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2.34 Joule, node 1 bernilai 2.31 Joule, node 2 bernilai 4.5 Joule, node 3 bernilai 5.9 Joule, node 4 bernilai 5.7 Joule, node 5 bernilai 6.1 Joule, node 6 bernilai 5.3 Joule, node 7 bernilai 6.1 Joule, node 8 bernilai 6.1 Joule, node 9 bernilai 7.3 Joule, node 10 bernilai 7.1 Joule., node 11 bernilai 6.2 Joule, node 12 bernilai 6.3 Joule, node 13 bernilai 6.4 Joule, node 14 bernilai 6.2 Joule, node 15 bernilai 5,24 Joule dan node 16 bernilai 5,34 Joule, node 17 bernilai 6.2 Joule, node 18 bernilai 6.4 Joule, node 19 bernilai 6.13 Joule, node 20 bernilai 6.4 Joule, node 21 bernilai 65.7 Joule dan node 45 bernilai 5.13 Joule.

Berdasarkan Gambar 4.7 dan Gambar 4.8 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 5.67 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 5.09 Joule.



Gambar 4.7. Energi End-to-end topologi mesh asumsi pertama routing DSDV



Gambar 4.8. Energi End-to-end topologi mesh asumsi pertama routing AODV

4.2.3. Topologi Tree

Desain topologi tree dapat dilihat pada Gambar 4.9 berikut ini.



Gambar 4.9. Topologi tree asumsi pertama

Terlihat pada Gambar 4.9 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya. Pada topologi ini hampir sama dengan topologi mesh, namun pada topologi ini komunikasi node terbagi atas beberapa jalur untuk sampai ke koordinator

Untuk sisi kiri ke kanan, jalur pertama yakni dari node 1 menuju node 3, node 5, node 7, node 9, node 11, node 13, node 15, node 14 baru ke node 45. Jalur kedua yakni node 0 menuju node 2, node 4, node 6, node 8, node 10, node 12, node 14, baru ke node 45. Jalur ke tiga dari node 17 ke node 19, node 21, node 20, node 18, node 16 baru ke node 45. Sedangkan sisi kanan ke kiri, jalur pertama yakni dari node 43 menuju node 41, node 39, node 37, node 35, node 33, node 13, node 31, node 29 baru ke node 44. Jalur kedua yakni node 42 menuju node 40, node 38, node 34, node 32, node 30, node 28, node 29, baru ke node 44. Jalur ke tiga dari node 26 ke node 24, node 22, node 23, node 25, node 27 baru ke node 44. Hasil dari simulasi tampak pada Tabel 4.5. dan Tabel 4.6. Hasil simulasi di ambil dari nilai rata-rata kedua sisi.

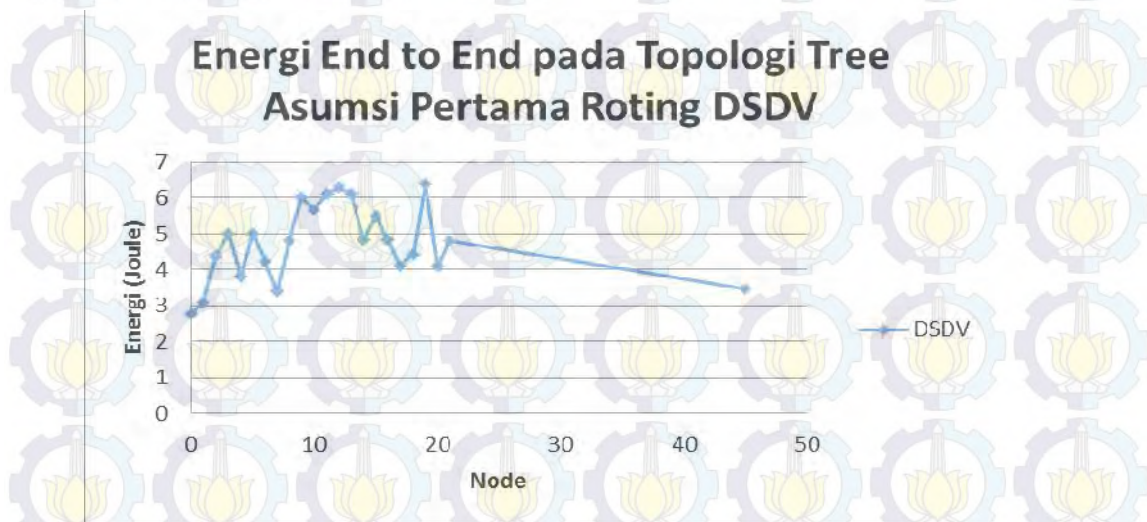
Tabel 4.5. Hasil simulasi topologi tree dengan routing DSDV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	6.35 kbps
2	<i>Delay end-to-end</i> rata-rata	26,34 ms
3	<i>Packet Loss</i>	2 paket
4	Energi <i>end-to-end</i> rata-rata	4.74 Joule

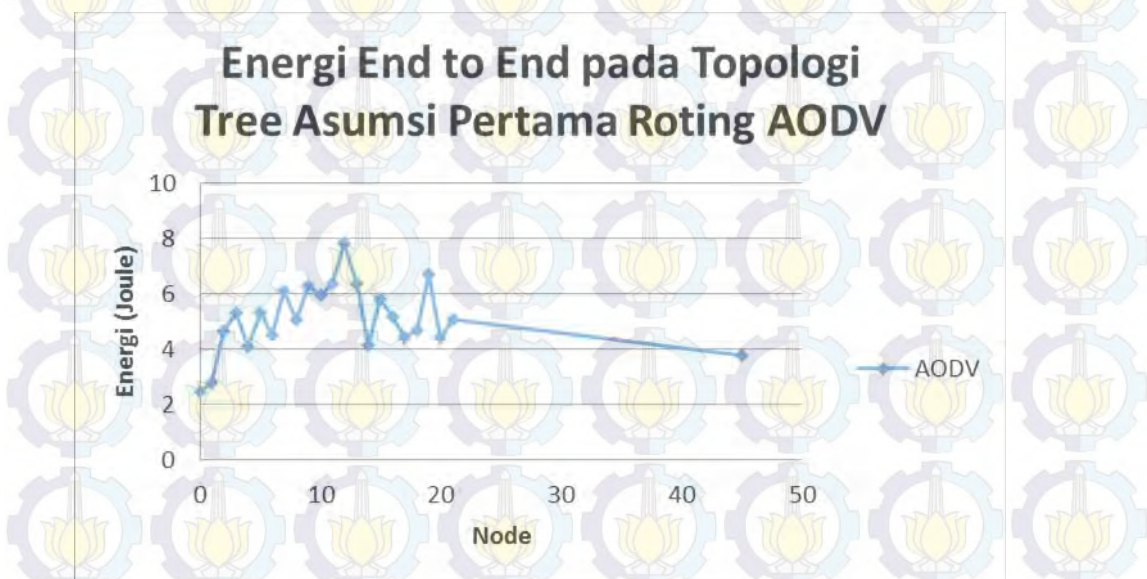
Tabel 4.6. Hasil simulasi topologi tree dengan routing AODV dengan asumsi pertama

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	7.56 kbps
2	<i>Delay end-to-end</i> rata-rata	19.50 ms
3	<i>Packet Loss</i>	9 paket
4	Energi <i>end-to-end</i> rata-rata	5.01 Joule

Untuk grafik energi end-to-end dapat dilihat pada Gambar 4.10 dan Gambar 4.11 berikut ini.



Gambar 4.10. Energi End-to-end topologi tree asumsi pertama routing DSDV



Gambar 4.11. Energi End-to-end topologi tree asumsi pertama routing AODV

Terlihat pada Gambar 4.10, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,48 Joule, node 1 bernilai 2,79 Joule, node 2 bernilai 4,67 Joule, node 3 bernilai 5,29 Joule, node 4 bernilai 4,12 Joule, node 5 bernilai 5,31 Joule, node 6 bernilai 4,32 Joule, node 7 bernilai 6,1 Joule, node 8 bernilai 5,1 Joule, node 9 bernilai 6,3 Joule, node 10

bernilai 5.96 Joule., node 11 bernilai 6.4 Joule, node 12 bernilai 7.82 Joule, node 13 bernilai 6.4 Joule, node 14 bernilai 4.15 Joule, node 15 bernilai 5.8 Joule dan node 16 bernilai 5.15 Joule, node 17 bernilai 4.41 Joule, node 18 bernilai 4.71 Joule, node 19 bernilai 6.7 Joule, node 20 bernilai 4.41 Joule, node 21 bernilai 5.1 Joule dan node 45 bernilai 3.78 Joule.

Terlihat pada Gambar 4.11, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,48 Joule, node 1 bernilai 2.79 Joule, node 2 bernilai 4.67 Joule, node 3 bernilai 5.29 Joule, node 4 bernilai 4.12 Joule, node 5 bernilai 5.31 Joule, node 6 bernilai 4.32 Joule, node 7 bernilai 6.1 Joule, node 8 bernilai 5.1 Joule, node 9 bernilai 6.3 Joule, node 10 bernilai 5.96 Joule., node 11 bernilai 6.4 Joule, node 12 bernilai 7.82 Joule, node 13 bernilai 6.4 Joule, node 14 bernilai 4.15 Joule, node 15 bernilai 5.8 Joule dan node 16 bernilai 5.15 Joule, node 17 bernilai 4.41 Joule, node 18 bernilai 4.71 Joule, node 19 bernilai 6.7 Joule, node 20 bernilai 4.41 Joule, node 21 bernilai 5.1 Joule dan node 45 bernilai 3.78 Joule.

Berdasarkan Gambar 4.9 dan Gambar 4.10 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 5.01 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 4.74 Joule.

Tabel 4.7 dan Tabel 4.8 merupakan tabel gabungan antara tabel gabungan topologi star, mesh, dan tree dengan asumsi pertama berdasarkan routingnya.

Tabel 4.7. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi pertama routing AODV

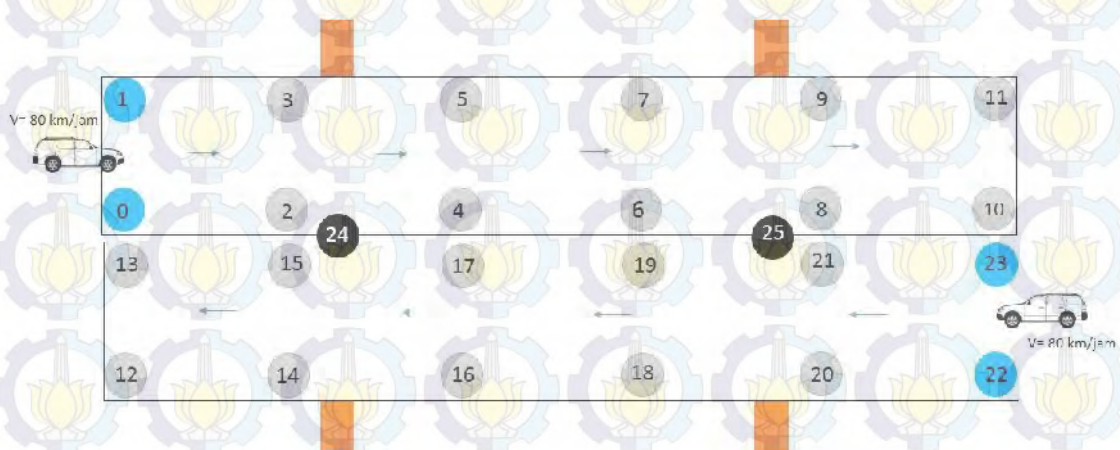
No.	Parameter Kinerja Jaringan	Topologi		
		Star	Mesh	Tree
1	<i>Throughput</i>	0.69 kbps	21.28 kbps	7.56 kbps
2	<i>Delay end-to-end</i> rata-rata	0.532 ms	10.68 ms	19.50 ms
3	<i>Packet Loss</i>	142 paket	1 paket	9 paket
4	Energi <i>end-to-end</i> rata-rata	0.154 Joule	5.66 Joule	5.01 Joule

Tabel 4.8. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi pertama routing DSDV

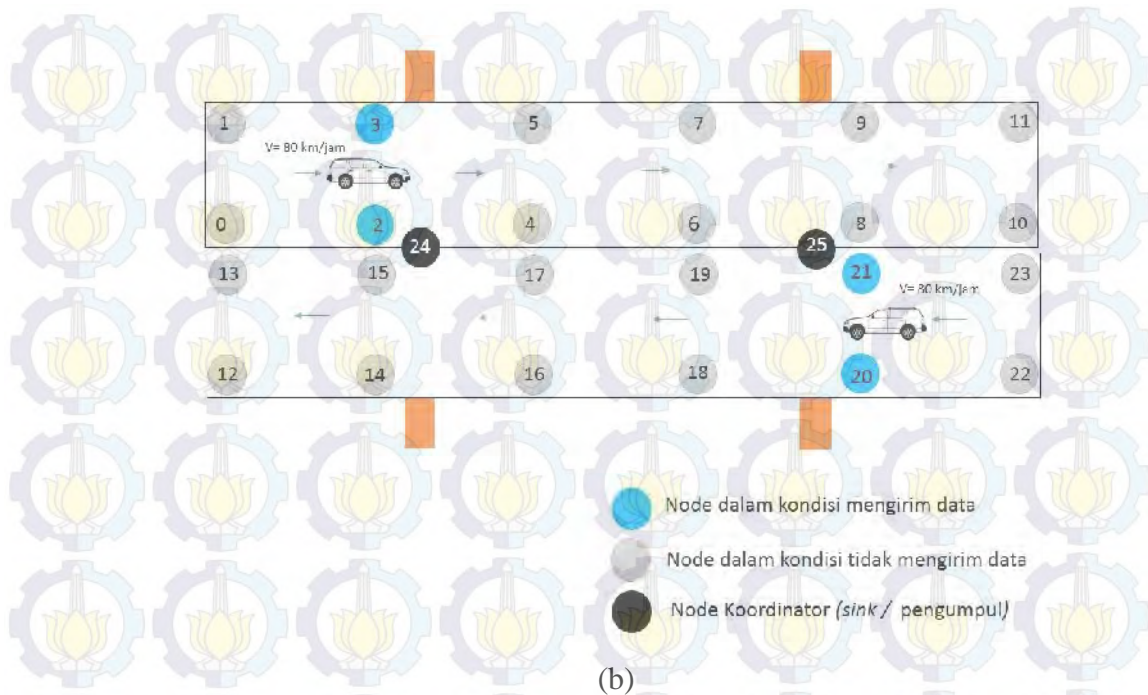
No.	Parameter Kinerja Jaringan	Topologi		
		Star	Mesh	Tree
1	Throughput	0.65 kbps	7.11 kbps	6.35 kbps
2	Delay end-to-end rata-rata	0.840 ms	27,25 ms	26,34 ms
3	Packet Loss	139 paket	2 paket	2 paket
4	Energi end-to-end rata-rata	0.106 Joule	5.09 Joule	4.74 Joule

4.3. Asumsi Kedua

Pada asumsi kedua ini, posisi node di kondisikan jarak antar node terletak dengan jarak 200 meter dan 12 meter. Skenario pada asumsi pertama yakni ketika terdapat sebuah kendaraan yang melintas dengan kecepatan 80 km/jam, sehingga node-node sensor akan mengikuti pergerakan kendaraan dan node yang akan dilewati akan bersiap-siap untuk mengirimkan paket data menuju node koordinator.



(a)

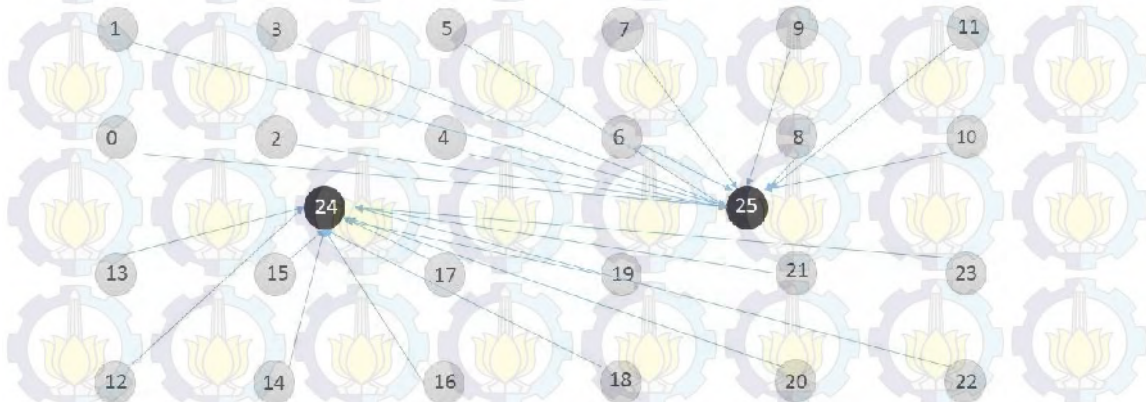


Gambar 4.12 Skenario pada asumsi kedua

Terlihat dari Gambar 4.12 dimana untuk skenario pada asumsi kedua. Pada sisi lajur kiri ke kanan ketika ada kendaraan melaju memasuki jembatan sehingga posisi kendaraan sama dengan node 0 dan node 1 maka node 0 dan node 1 akan mengirim paket data menuju node 25 (node koordinator). Sedangkan pada sisi lajur kanan ke kiri ketika ada kendaraan melaju memasuki jembatan maka node 22 dan node 23 akan mengirim paket data menuju node 24 (node koordinator) seperti pada Gambar 4.12 (a). Node aktif mengirimkan data ditandai dengan node berwarna biru. Pada sisi lajur kiri ke kanan ketika posisi kendaraan sama dengan node 2 dan node 3 maka node 2 dan node 2 akan mengirim paket data menuju node 25, untuk node 0 dan node 1 sudah tidak lagi mengirimkan paket data. Sedangkan Pada sisi lajur kanan ke kiri ketika posisi kendaraan sama dengan node 20 dan node 21 maka node 20 dan node 21 akan mengirim paket data menuju node 24, untuk node 22 dan node 23 sudah tidak lagi mengirimkan paket data seperti pada Gambar 4.12 (b).

4.3.1. Topologi Star

Topologi star merupakan topologi paling sederhana dimana komunikasi terjadi secara *point-to-point* node dengan sink. Desain topologi seperti Gambar 4.11.



Gambar 4.13. Topologi star asumsi pertama

Terlihat pada Gambar 4.13, bahwa pada topologi star setiap node berkomunikasi secara langsung dengan node koordinator (node 24 dan node 25).

Topologi pada asumsi yang ini mirip dengan Gambar 4.13 dimana setiap node mengirim paket data langsung menuju koordinator. Pada sisi kiri ke kanan awalnya, node 0 dan node 1 akan mengirimkan paket data secara bersamaan dan langsung menuju node 25 selama 9 detik, kemudian node 2 dan node 3 akan mengirimkan paket data secara bersamaan dan langsung menuju node 25 selama 9 detik. Dilanjutkan node 4 dan node 5 mengirimkan paket data, lalu node 6 dan node 7, dan yang terakhir adalah node 10 dan node 11 mengirimkan paket data secara bersamaan langsung menuju node 25. Sedangkan Pada sisi kanan ke kiri awalnya, node 22 dan node 23 akan mengirimkan paket data secara bersamaan dan langsung menuju node 24 selama 9 detik, kemudian node 20 dan node 21 akan mengirimkan paket data secara bersamaan dan langsung menuju node 24 selama 9 detik. Dilanjutkan node 18 dan node 19 mengirimkan paket data, lalu node 16 dan node 17, dan yang terakhir adalah node 12 dan node 13 mengirimkan paket data secara bersamaan langsung menuju node 24. Hasil dari simulasi

tampak pada Tabel 4.9. dan Tabel 4.10. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

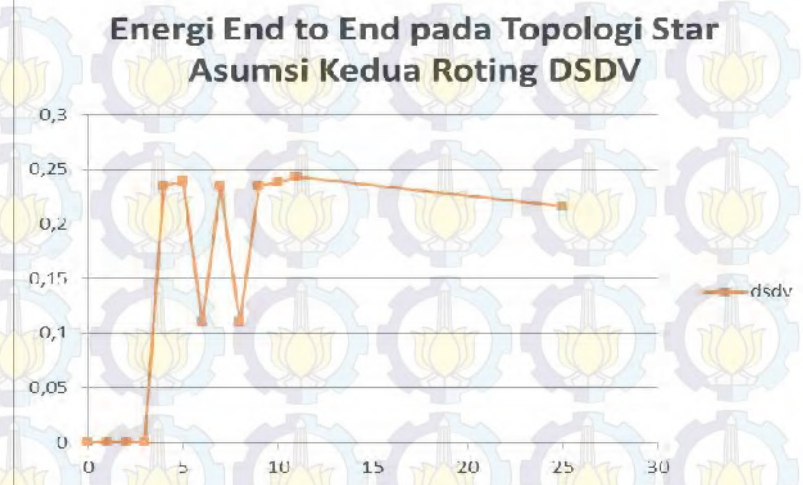
Tabel 4.9. Hasil simulasi topologi star dengan routing DSDV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	0.55 kbps
2	<i>Delay end-to-end</i> rata-rata	0.940 ms
3	<i>Packet Loss</i>	152 paket
4	Energi <i>end-to-end</i> rata-rata	0.143 Joule

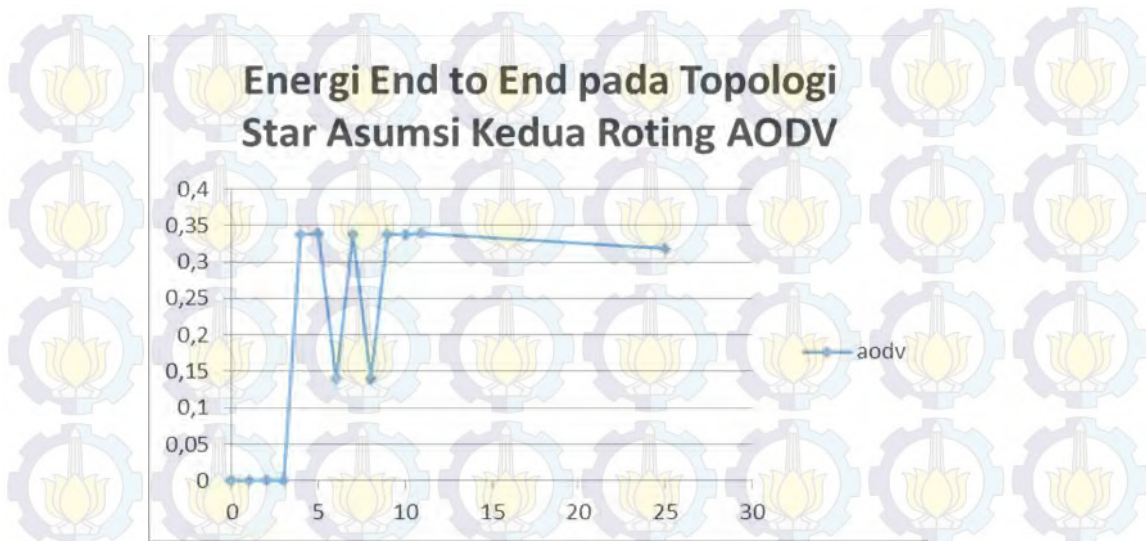
Tabel 4.10. Hasil simulasi topologi star dengan routing AODV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	0.48 kbps
2	<i>Delay end-to-end</i> rata-rata	0.199 ms
3	<i>Packet Loss</i>	149 paket
4	Energi <i>end-to-end</i> rata-rata	0.202 Joule

karena terdapat node yang melebihi jarak jangkauan maka terdapat beberapa paket yang tidak dikirimkan.



Gambar 4.14. Energi End-to-end topologi star asumsi kedua routing DSDV



Gambar 4.15. Energi End-to-end topologi star asumsi kedua routing AODV

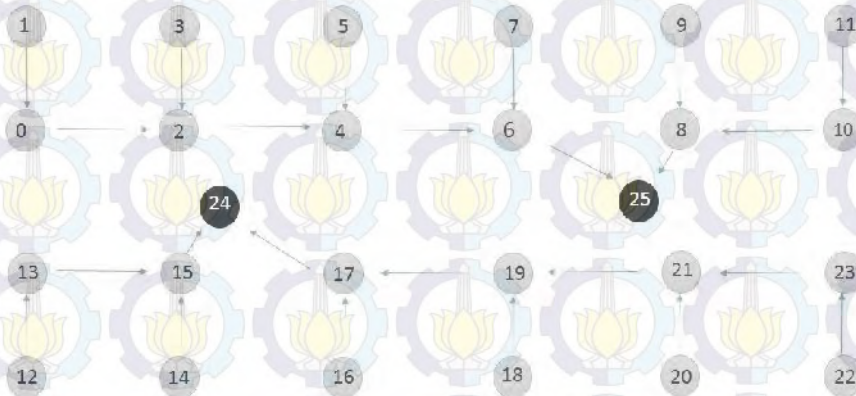
Parameter energi *end-to-end* pada topologi star asumsi pertama untuk routing AODV dan DSDV pada masing-masing node ditunjukkan pada Gambar 4.14 dan Gambar 4.15. Pada lajur kanan ke kiri untuk routing AODV node 0, node 1, node 2 dan node 3, energi *end-to-end* masing-masing node tersebut adalah nol. Hal ini disebabkan ketika node-node tersebut mengirimkan paket data secara langsung menuju node 0, karena jarak yang melebihi jangkauan XBee maka tidak paket tersebut tidak dikirimkan. Sedangkan pada node 4, node 5, node 7, node 8, node 9, node 10 dan node 11 masing-masing memiliki nilai energi *end-to-end* untuk routing AODV berturut-turut 0,338 Joule, 0,339 Joule, 0,338 Joule, 0,338 Joule, 0,338 Joule, dan 0,339 Joule. Node – node ini untuk aktifitas “mendengar”. Node 6 dan node 8 memiliki nilai energi *end-to-end* 0,14 Joule untuk aktifitas pengiriman paket data menuju node 25 (Koordinator). Sedangkan node 25 memiliki nilai energi *end-to-end* 0,318 Joule untuk aktifitas menerima paket data dan “mendengar. Sedangkan untuk routing DSDV berturut-turut 0,235 Joule, 0,239 Joule, 0,235 Joule, 0,235 Joule, 0,238 Joule dan 0,243 Joule. Node 6 dan node 7 memiliki nilai energi *end-to-end* 0,11 untuk aktifitas pengiriman paket data menuju node 25 (Koordinator). Sedangkan node 25 memiliki nilai energi *end-to-end* 0,216 Joule untuk aktifitas menerima paket data dan “mendengar.

Berdasarkan Gambar 4.14 dan Gambar 4.15 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 0,202 Joule.

Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 0,143 Joule.

4.3.2. Topologi Mesh

Topologi Mesh atau biasa disebut komunikasi *peer-to-peer* adalah komunikasi yang terjadi antar node Desain topologi seperti Gambar 4.16.



Gambar 4.16. Topologi mesh asumsi kedua

Terlihat pada Gambar 4.16 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya. Pada skenario dengan asumsi pertama ini, pada sisi kiri ke kanan, ketika sebuah mobil memiliki koordinat yang sama dengan node 1 dan node 0, maka node 1 mengirimkan data melalui node 0 yang kemudian di teruskan dari node 0 menuju node 2. Node 2 juga menerima data dari node 3 dan seterusnya sehingga data sampai di node 25. Sedangkan pada sisi kanan ke kiri, ketika sebuah mobil memiliki koordinat yang sama dengan node 22 dan node 23, maka node 22 mengirimkan data melalui node 23 yang kemudian di teruskan dari node 23 menuju node 21. Node 21 juga menerima data dari node 20 dan seterusnya sehingga data sampai di node 24.

Hasil dari simulasi tampak pada Tabel 4.11. dan Tabel 4.12. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

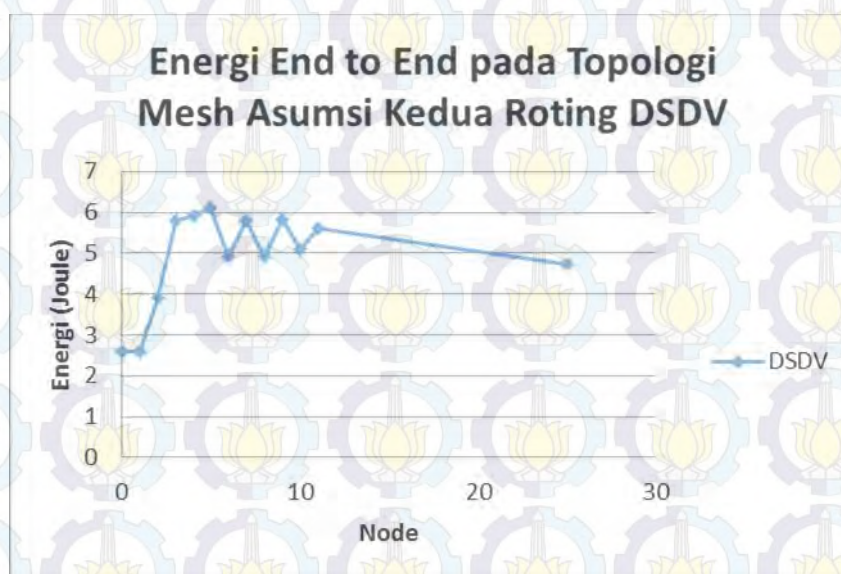
Tabel 4.11. Hasil simulasi topologi mesh dengan routing DSDV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	5.35 kbps
2	<i>Delay end-to-end rata-rata</i>	27.45 ms
3	<i>Packet Loss</i>	4 paket
4	Energi <i>end-to-end</i> rata-rata	4.91 Joule

Tabel 4.12. Hasil simulasi topologi mesh dengan routing AODV dengan asumsi kedua

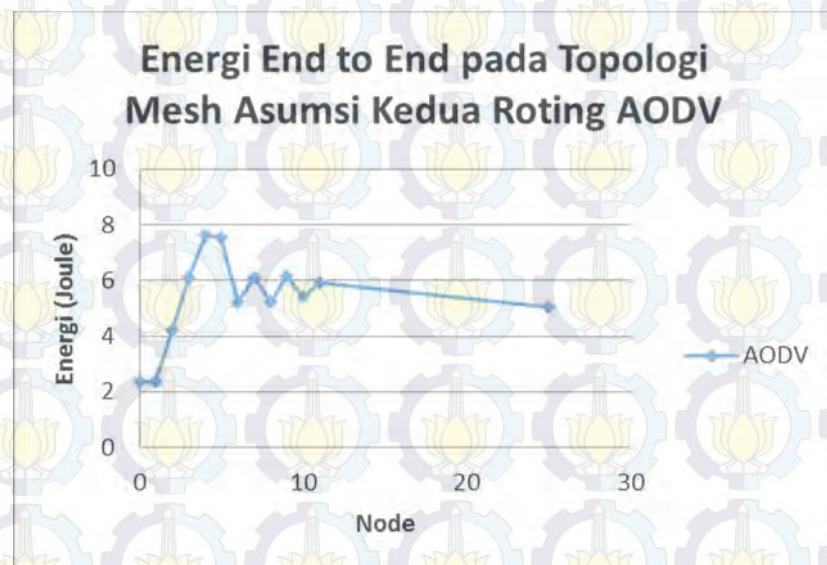
No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	19.89 kbps
2	<i>Delay end-to-end rata-rata</i>	19.65 ms
3	<i>Packet Loss</i>	2 paket
4	Energi <i>end-to-end</i> rata-rata	5.32 Joule

Untuk grafik energi end-to-end dapat dilihat pada Gambar 4.17 dan Gambar 4.18 berikut ini.



Gambar 4.17. Energi End-to-end topologi mesh asumsi kedua routing DSDV

Terlihat pada Gambar 4.17, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2.58 Joule, node 1 bernilai 2.58 Joule, node 2 bernilai 3.91 Joule, node 3 bernilai 5.8 Joule, node 4 bernilai 5.93 Joule, node 5 bernilai 6.12 Joule, node 6 bernilai 4.93 Joule, node 7 bernilai 4.8 Joule, node 8 bernilai 5.93 Joule, node 9 bernilai 5.83 Joule, node 10 bernilai 5.1 Joule dan node 11 bernilai 5.6 Joule, sedangkan node 25 bernilai 4.72 Joule.



Gambar 4.18. Energi End-to-end topologi mesh asumsi kedua routing AODV

Terlihat pada Gambar 4.18, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2.38 Joule, node 1 bernilai 2.38 Joule, node 2 bernilai 4.31 Joule, node 3 bernilai 6.1 Joule, node 4 bernilai 7.63 Joule, node 5 bernilai 7.52 Joule, node 6 bernilai 5.23 Joule, node 7 bernilai 6.1 Joule, node 8 bernilai 5.23 Joule, node 9 bernilai 6.13 Joule, node 10 bernilai 5.4 Joule dan node 11 bernilai 5.9 Joule, sedangkan node 25 bernilai 5.02 Joule.

Berdasarkan Gambar 4.17 dan Gambar 4.18 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 5.32 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 4.91 Joule.

4.3.3. Topologi Tree

Desain topologi tree dapat dilihat pada Gambar 4.19 berikut ini.



Gambar 4.19. Topologi tree asumsi kedua

Terlihat pada Gambar 4.19 setiap node terhubung satu sama lain dan dapat mengirimkan data kepada node tetangganya. Pada topologi ini hampir sama dengan topologi mesh, namun pada topologi ini komunikasi node terbagi atas beberapa jalur untuk sampai ke koordinator.

Tabel 4.13. Hasil simulasi topologi tree dengan routing DSDV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	5.71 kbps
2	<i>Delay end-to-end</i> rata-rata	25.97 ms
3	<i>Packet Loss</i>	5 paket
4	Energi <i>end-to-end</i> rata-rata	4.64 Joule

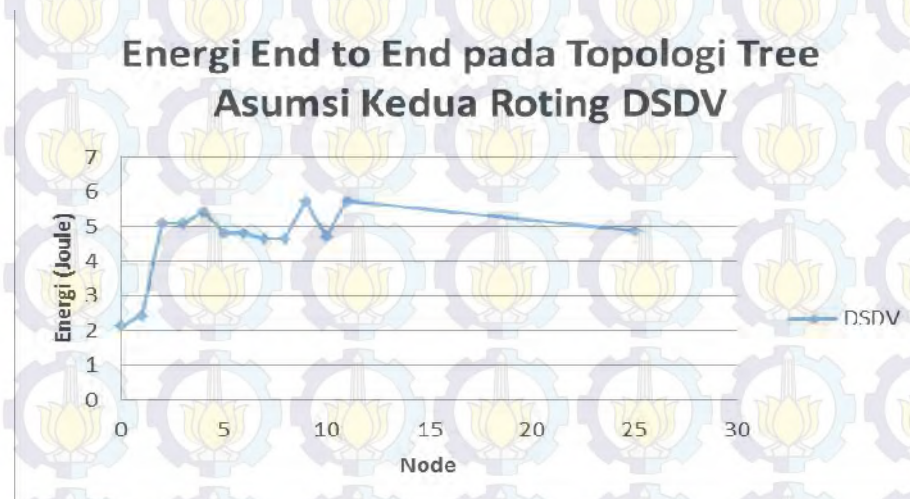
Untuk sisi kiri ke kanan, jalur pertama yakni dari node 1 menuju node 3, node 5, node 7, node 6 baru ke node 25. Jalur kedua yakni node 0 menuju node 2, node 4, node 6 baru ke node 25. Jalur ke tiga dari node 9 ke node 11, node 10, node 8 baru ke node 25. Sedangkan sisi kanan ke kiri, jalur pertama yakni dari

node 23 menuju node 21, node 19, node 17 baru ke node 24. Jalur kedua yakni node 22 menuju node 20, node 18, node 16 baru ke node 24. Jalur ke tiga dari node 14 ke node 12, node 13, node 15 baru ke node 24. Hasil dari simulasi tampak pada Tabel 4.13. dan Tabel 4.14. Hasil simulasi diambil dari nilai rata-rata kedua sisi.

Tabel 4.14. Hasil simulasi topologi tree dengan routing AODV dengan asumsi kedua

No.	Parameter Kinerja Jaringan	Nilai
1	<i>Throughput</i>	6.98 kbps
2	<i>Delay end-to-end</i> rata-rata	20.43 ms
3	<i>Packet Loss</i>	8 paket
4	Energi <i>end-to-end</i> rata-rata	4.52 Joule

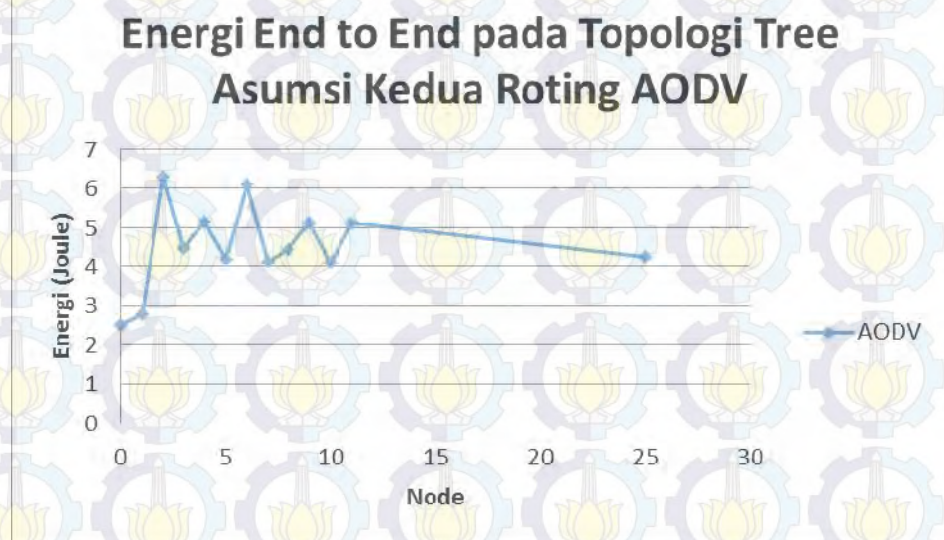
Untuk grafik energi end-to-end dapat dilihat pada Gambar 4.20 dan Gambar 4.21 berikut ini.



Gambar 4.20. Energi End-to-end topologi tree asumsi kedua routing DSDV

Terlihat pada Gambar 4.20, Untuk routing DSDV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,12 Joule, node 1

bernilai 2.41 Joule, node 2 bernilai 5.1 Joule, node 3 bernilai 5.08 Joule, node 4 bernilai 5.4 Joule, node 5 bernilai 4.81 Joule, node 6 bernilai 4.81 Joule, node 7 bernilai 4.65 Joule, node 8 bernilai 4.65 Joule, node 9 bernilai 5.73 Joule, node 10 bernilai 4.7 Joule., node 11 bernilai 5.73 Joule. Sedangkan node 25 bernilai 4.67 Joule.



Gambar 4.21. Energi End-to-end topologi tree asumsi kedua routing AODV

Terlihat pada Gambar 4.21, Untuk routing AODV masing-masing node memiliki nilai energi *end-to-end* yaitu untuk node 0 bernilai 2,52 Joule, node 1 bernilai 2.81 Joule, node 2 bernilai 6.3 Joule, node 3 bernilai 4.48 Joule, node 4 bernilai 5.15 Joule, node 5 bernilai 4.2 Joule, node 6 bernilai 6.1Joule, node 7 bernilai 4.15 Joule, node 8 bernilai 4.45 Joule, node 9 bernilai 5.13 Joule, node 10 bernilai 4.1 Joule., node 11 bernilai 5.13 Joule. Sedangkan node 25 bernilai 4.27 Joule.

Berdasarkan Gambar 4.20 dan Gambar 4.21 tersebut didapatkan energi *end-to-end* untuk routing AODV rata-rata komunikasi adalah 4.523 Joule. Sedangkan energi *end-to-end* untuk routing DSDV rata-rata komunikasi adalah 4.64 Joule

Tabel 4.15 dan Tabel 4.16 merupakan tabel gabungan antara tabel gabungan topologi star, mesh, dan tree dengan asumsi kedua berdasarkan routingnya.

Tabel 4.15. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi kedua routing AODV

No.	Parameter Kinerja	Topologi		
	Jaringan	Star	Mesh	Tree
1)	<i>Throughput</i>	0.48 kbps	19.89 kbps	6.98 kbps
2)	<i>Delay end-to-end</i> rata-rata	0.199 ms	19.65 ms	20.43 ms
3)	<i>Packet Loss</i>	149 paket	2 paket	8 paket
4)	Energi <i>end-to-end</i> rata-rata	0.202 Joule	5.32 Joule	4.52 Joule

Tabel 4.16. Tabel gabungan hasil simulasi topologi star, mesh dan tree asumsi kedua routing DSDV

No.	Parameter Kinerja	Topologi		
	Jaringan	Star	Mesh	Tree
1	<i>Throughput</i>	0.55 kbps	5.35 kbps	6.98 kbps
2	<i>Delay end-to-end</i> rata-rata	0.940 ms	27.45 ms	20.43 ms
3	<i>Packet Loss</i>	152 paket	4 paket	8 paket
4	Energi <i>end-to-end</i> rata-rata	0.143 Joule	4.91 Joule	4.52 Joule

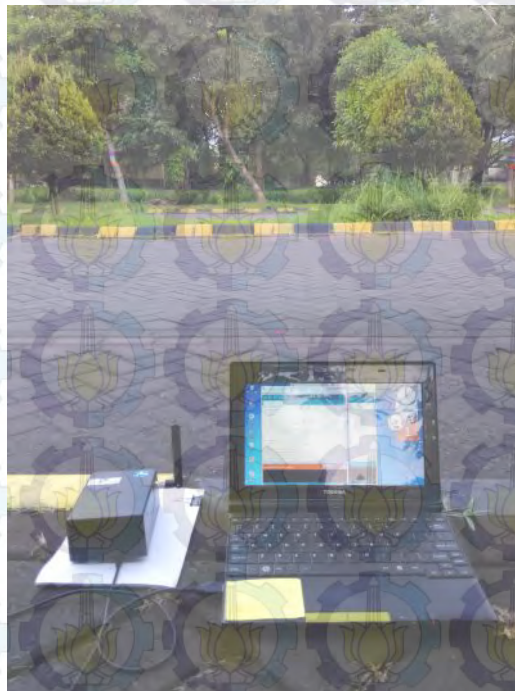
4.4. Analisa

Dari hasil yang didapat terlihat bahwa antara asumsi pertama memiliki nilai-nilai kinerja yang baik khususnya pada topologi mesh dengan routing AODV namun pada asumsi ini memerlukan banyak node. Sedangkan pada asumsi kedua nilai-nilai kinerja cukup baik. Sebanding dengan jarak yang semakin jauh maka nilai *throughput* semakin kecil. Pada asumsi ini memiliki kelebihan node

yang sedikit sehingga mampu meminimalisir jumlah node, sehingga mampu lebih meminimalisir energi yang diperlukan.

4.5. Validasi Parameter Simulasi

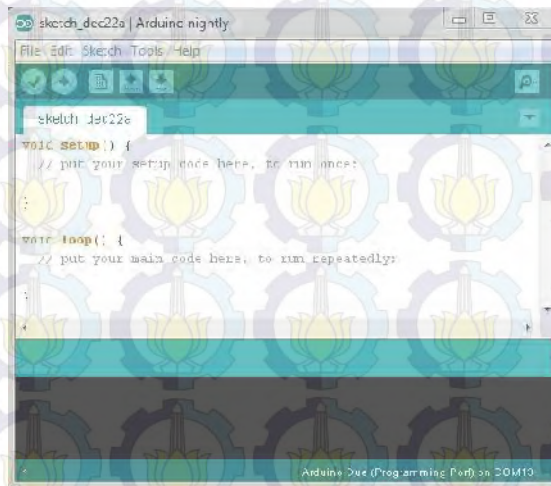
Pada proses validasi parameter simulasi ini dilakukan pengujian untuk parameter yang digunakan dalam simulasi. Pengujian ini dengan mengirimkan data sensor accelerometer kemudian melihat apakah data tersebut dapat diterima oleh node koordinator/*sink* dengan baik atau tidak. Selanjutnya juga menghitung *packet loss*, *delay*, *throughput* dan konsumsi arus yang digunakan dari baterai. Setelah menetapkan terlebih dahulu posisi node yang di bagi dalam dua asumsi yakni asumsi pertama jarak antara koordinator ke sensor 1 yakni 12 meter sama dengan jarak sensor 3 ke sensor 2, dan sensor 2 ke koordinator 100 meter sama dengan jarak sensor 3 ke sensor 1. Sedangkan asumsi kedua jarak antara koordinator ke sensor 1 yakni 12 meter sama dengan jarak sensor 3 ke sensor 2, sedangkan sensor 2 ke koordinator 200 meter sama dengan jarak sensor 3 ke sensor 1.



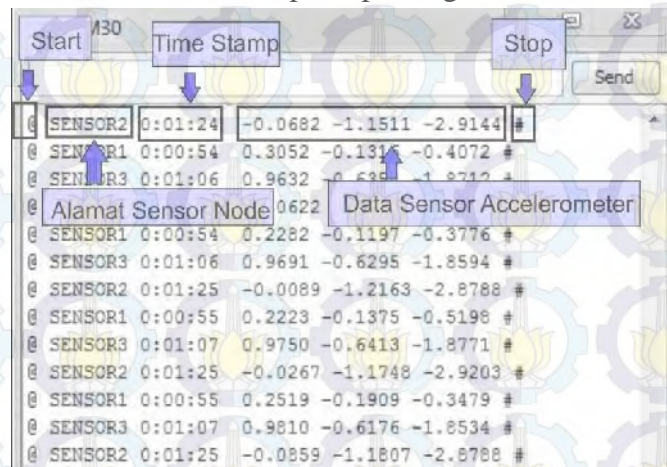
Gambar 4.22. Proses pengambilan data secara real

4.6. Pengujian Kalibrasi Node

Pengujian dilakukan pada node koordinator. Pengujian dilakukan untuk mendapatkan data pembacaan sensor accelerometer sebelum dikirimkan pada node koordinator. Berikut Gambar 4.24 menunjukkan data yang diterima oleh penerima melalui port serial yang terhubung dengan desktop ataupun notebook.



Gambar 4.23 Tampilan pemrograman arduino



Gambar 4.24 window serial monitoring pada node koordinator

4.7. Manajemen Transmisi Data

Proses manajemen transmisi data pada penelitian ini model komunikasi berdasarkan terjadinya suatu kejadian. Node yang telah dibagi dalam 2 macam yakni node sensor dan node koordinator/sink. Node sensor memiliki fungsi sebagai sensing node yang akan mensensing adanya getaran yang tertangkap di daerah sensing node. Node sensor yang berada paling ujung (*End device*) melanjutkan pengiriman datanya melalui node sensor terdekatnya. Node sensor

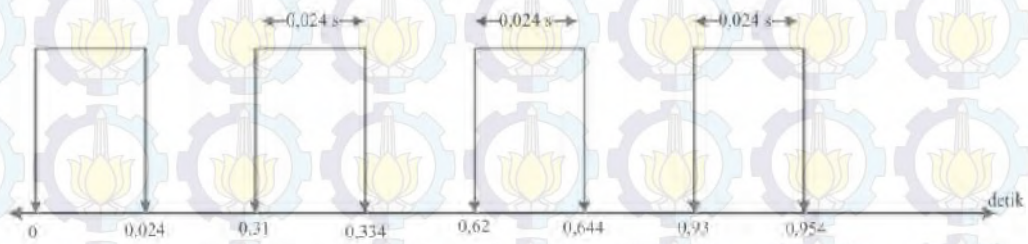
yang berada dekat dengan node koordinator (router) akan menerima data dari node sensor sebelumnya dan data hasil pembacaan sensor dikirimkan pada node koordinator. Sedangkan node koordinator berfungsi sebagai pengumpul data sensing dari node sensor yang kemudian data akan dikirimkan pada base station. sensor sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *awake* (node aktif), Sedangkan ketika node sensor tidak sedang melakukan proses penginderaan berarti node ini berada dalam kondisi *sleep*. *Sleep* bukan berarti node mati namun node berada pada kondisi *idle* atau hanya mendengar saja. Pada tahap penginderaan yang dilakukan pada penelitian ini adalah kalibrasi dari sensor memastikan sensor yang digunakan dapat berfungsi dengan baik. Berikut pengujian dari proses transmisi data sensor accelerometer yang digunakan pada node jaringan sensor di sistem monitoring kesehatan struktur ini. Dari penelitian yang dilakukan, desain terbaik terdapat pada topologi mesh dengan desain seperti pada Gambar 4.14. Disini ketika node 1 akan mengirim data, melewati node 2 kemudian dari node 2 mengirimkan data ke node koordinator. Kemudian dilanjutkan ketika node 0 akan mengirim data ke node 2 kemudian dari node 2 akan mengirim ke node koordinator. Penentuan node selanjutnya berdasarkan node terdekat dengan jarak terdekat berdasarkan route terdekat dan jumlah energi dari setiap node. Ketika node akan berlanjut ke node selanjutnya node melakukan pengecekan terhadap energinya ketika energi cukup maka proses dilanjutkan. Begitu seterusnya sampai semua node terlewati. Pada topologi jaringan mesh dicover oleh dua buah sink node 24 dan 25. Cluster pertama dicover oleh node 24 dengan node 0,1,2,3,4,5,12,13,14,15,16 dan 17. Kemudian cluster kedua dicover oleh node 25 dengan node 6,7,8,9,10,11,18,19,20,21,22 dan 23. Jadi setiap koordinator mengcover 12 node dan 1 node koordinator.

Sedangkan dalam proses pentransmisi data besar byte yang dikirim adalah 12 byte dengan perincian sesuai Gambar 4.25 berikut :

Start byte (1 byte)	Alamat Sensor dan Jenis Sensor (2 byte)	Data Monitoring (8 byte)	Stop byte (1 byte)
------------------------	--	-----------------------------	-----------------------

Gambar 4.25 Format Protokol Pemaketan Data pada Node Sensor

Pada proses pengiriman data setiap paket berisi 12 Byte. Sehingga ketika pengiriman data dengan panjang byte 1000 byte terkirim dalam jangka 2 detik. jadi dapat disimpulkan bahwa setiap 1 paket dengan panjang 12 byte terkirim dalam jangka waktu 0,024 detik. Ilustrasi pengiriman dapat dilihat pada Gambar 4.26.



Gambar 4.26 Waktu Pengiriman data

Dari hasil yang diujikan dapat disimpulkan bahwa node sensor dapat menerima pengindraan getaran dari kendaraan yang melintas dan mengirimkan langsung secara singlehop dan dihitung berapakah konsumsi arus dari satu node sensor ini selama satu jam pengujian. Pada kondisi LOS node sensor dicatu oleh baterai. Dengan kapasitas 220 mA dalam waktu satu jam pengukuran. Pengukuran berlangsung ketika siang hari kondisi jalanan dengan intensitas kendaraan cukup ramai. Pada awal pengukuran baterai diukur dahulu dalam kondisi close circuit voltage dengan multimeter sebesar 9,6 Volt. Setelah satu jam pengukuran, diukur kembali pada baterai untuk mengetahui berapakah yang dibutuhkan oleh node sensor satu jam pengukuran dalam kondisi trafik jalanan yang cukup padat. Kemudian besar tegangan baterai sebesar 8,5 Volt ini saat kondisi node *awake*, sedangkan pada kondisi node *sleep (idle)* yakni kondisi dimana sedang tidak ada kendaraan yang lewat, besar tegangan setelah 1 jam yakni 8,7 Volt. Jadi dapat disimpulkan bahwa satu jam tegangan yang digunakan saat kondisi node *awake* (aktif) sebesar 1,1 Volt, sedangkan node *sleep (idle)* sebesar 0.9 Volt. Perhitungan besar konsumsi arus yang diperlukan pada kondisi node *awake* (aktif) dan node *sleep (idle)* berdasarkan rumus berikut ini :

$$\frac{\text{Besar Tegangan Terukur}}{\text{Isi Baterai}} \times 100 \times 0.22Ah = 4.1$$

Karena node dicatu oleh baterai dengan *brand* Imedion berkapasitas 9.6 V dengan kapasitas arus 220 mA atau sama dengan 0.22 Ah. Maka untuk Isi baterai sebesar 9.6 V. Kemudian data dimasukkan pada rumus diatas untuk mendapat konsumsi arus selama satu jam. Berikut perhitungannya:

- Kondisi node *awake* (aktif)

$$\frac{1,1}{9,6} \times 100 \times 0.22 \text{Ah} = 2,52 \text{Ah}$$

- Kondisi node *sleep* (*idle*)

$$\frac{0,9}{9,6} \times 100 \times 0.22 \text{Ah} = 2,06 \text{Ah}$$

Didapatkan kondisi bahwa pada saat node *awake* (aktif) membutuhkan konsumsi arus sebesar 2.52 Ah sedangkan saat kondisi node *sleep* (*idle*) membutuhkan konsumsi arus sebesar 2.06 Ah selama satu jam pengukuran. Berikut pada Tabel 4.17 menunjukkan perbandingan konsumsi arus dari kedua kondisi node.

Tabel 4.17 Perbandingan Konsumsi Arus Berdasarkan Kondisi Node

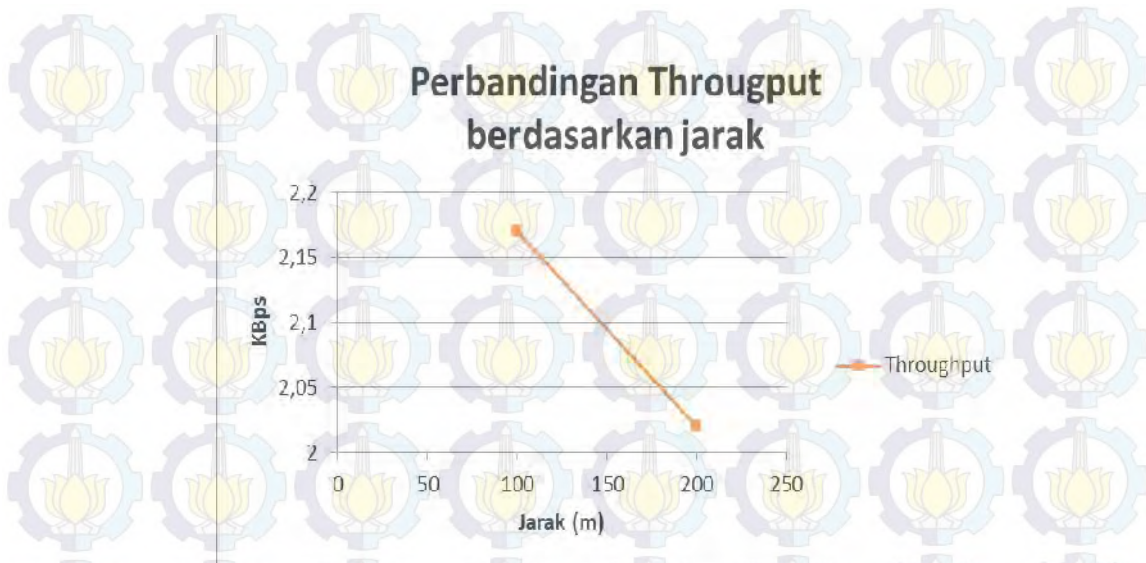
Kondisi Node	Tegangan (V)	Konsumsi Arus (Ah)
Sleep	0.9	2.06
Awake	1.1	2.52

4.8. Analisa Validasi Parameter Simulasi

Pada proses validasi parameter simulasi ini didapatkan hasil perbandingan antara jarak 100 meter dengan jarak 200 meter. Didapatkan hasil seperti pada Tabel 4.18 berikut ini :

Tabel 4.18 Hasil Pengukuran *Throughput*

Jarak	Paket data yang diterima dalam 1 menit	<i>Throughput</i> node (KBps)
	node (byte)	
100 m	130300	2.17
200 m	121100	2.02



Gambar 4.27 Perbandingan throughput validasi

Pada pengukuran *throughput* Tabel 4.13 dengan jarak 100 meter pada node sensor, node koordinator menerima data paket sebesar 130300 Byte dari node dengan persamaan 3.6 diperoleh nilai *throughput* sebesar 2.17 KBps, sedangkan untuk jarak 200 m pada node sensor, node koordinator menerima data paket sebesar 121100 Byte dari node diperoleh nilai *throughput* sebesar 2.02 KBps. Terlihat pada Tabel 4.13 mengalami penurunan sekitar 0.15 KBps. Hal ini disebabkan banyaknya paket data yang hilang karena pengaruh propagasi, dengan jarak yang semakin jauh membutuhkan waktu propagasi yang besar/*delay*, sehingga paket yang diterima dalam 1 menit berkurang.

Sementara itu dalam Pengujian *delay* disini dilakukan untuk mengetahui kinerja sistem terhadap pengaruh jarak terhadap lama waktu pengiriman (end to end delay). Pada pengujian delay ini peneliti mengirimkan data sebesar 100 byte, kemudian dalam pengiriman data tersebut jarak diubah berdasarkan parameter simulasi, hal ini untuk mengetahui pengaruh jarak terhadap lama pengiriman dengan melihat time stamp yang ditunjukkan pada saat penerimaan data di node koordinator. Pengukuran delay yang terjadi pada node sensor dapat dilihat pada Tabel 4.19. Gambar 4.25 menunjukkan grafik pengaruh jarak terhadap delay penerimaan data pada node koordinator.

Tabel 4.19 Hasil Pengukuran Delay

Jarak	Delay node (ms)
100 m	39
200 m	67



Gambar 4.28 Perbandingan delay validasi

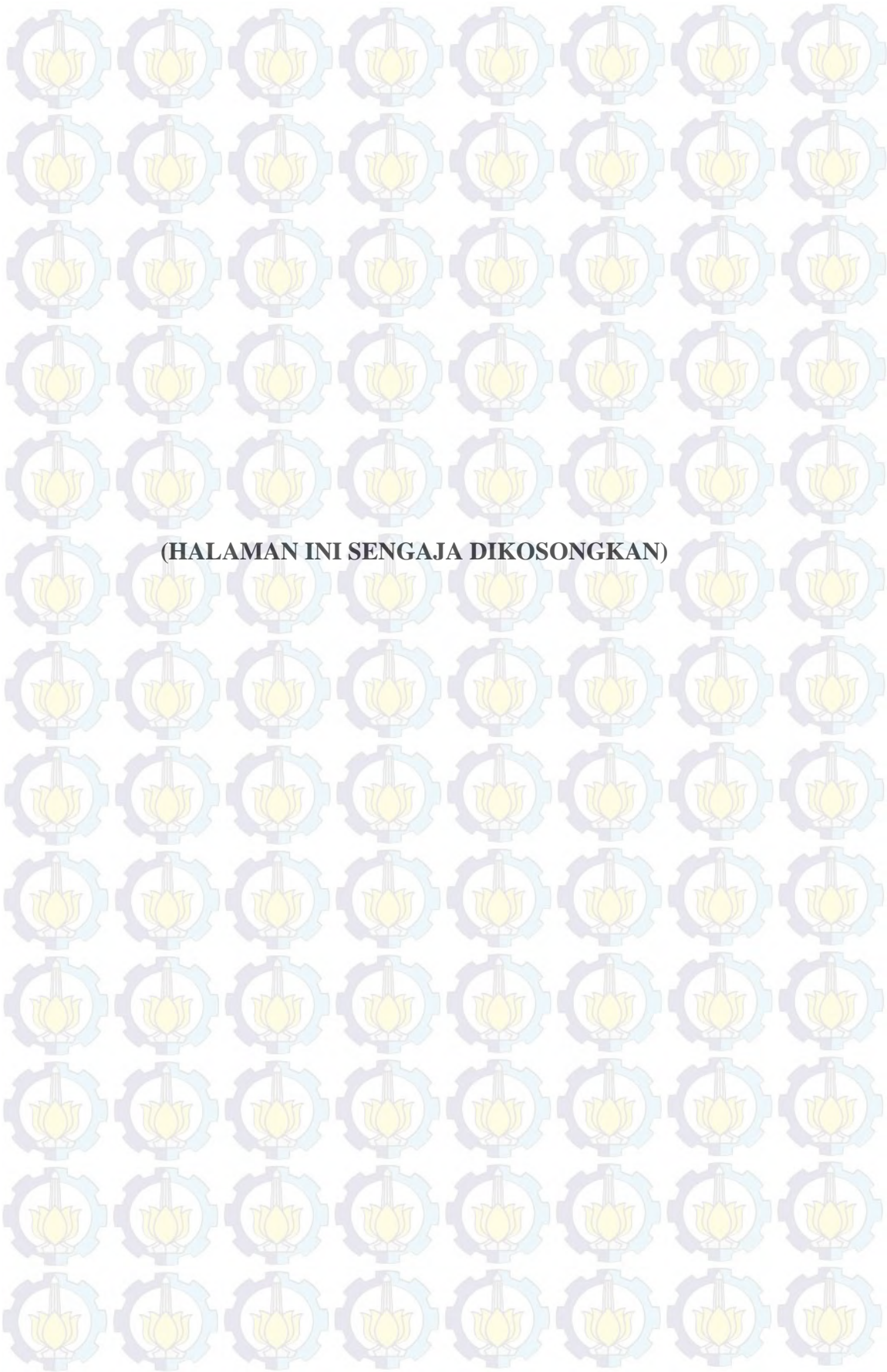
Untuk konsumsi arus yang digunakan pengamatan dilakukan dengan dua macam rancangan dimana rancangan pertama konsumsi arus di hitung dengan kondisi node tanpa bebas dan rancangan kedua kondisi node dengan beban sensor. Pada setiap node dicatu oleh baterai dengan *brand* Imedion berkapasitas 9.6 V. Pada kondisi baterai terisi penuh kapasitas arus 220 mA kemudian ketika diukur arusnya setiap 30 menit maka di dapatkan hasil sebagai berikut.

Tabel 4.20 Hasil pengukuran konsumsi arus

Kondisi	Sisa kapasitas saat pengukuran 30 menit pertama	Konsumsi rata-rata tiap 30 menit	Waktu habis
Tanpa Beban	207.1 mA	13 mA	8 Jam
Dengan Beban	194.3 mA	26 mA	4 Jam

Terlihat dari Tabel 4.20 bahwa untuk kondisi tanpa beban pada 30 menit pertama sisa kapasitas arus sebesar 207.1 mA sehingga rata-rata konsumsi arus tiap 30 menit akan berkurang sebanyak 13 mA maka baterai akan habis dalam

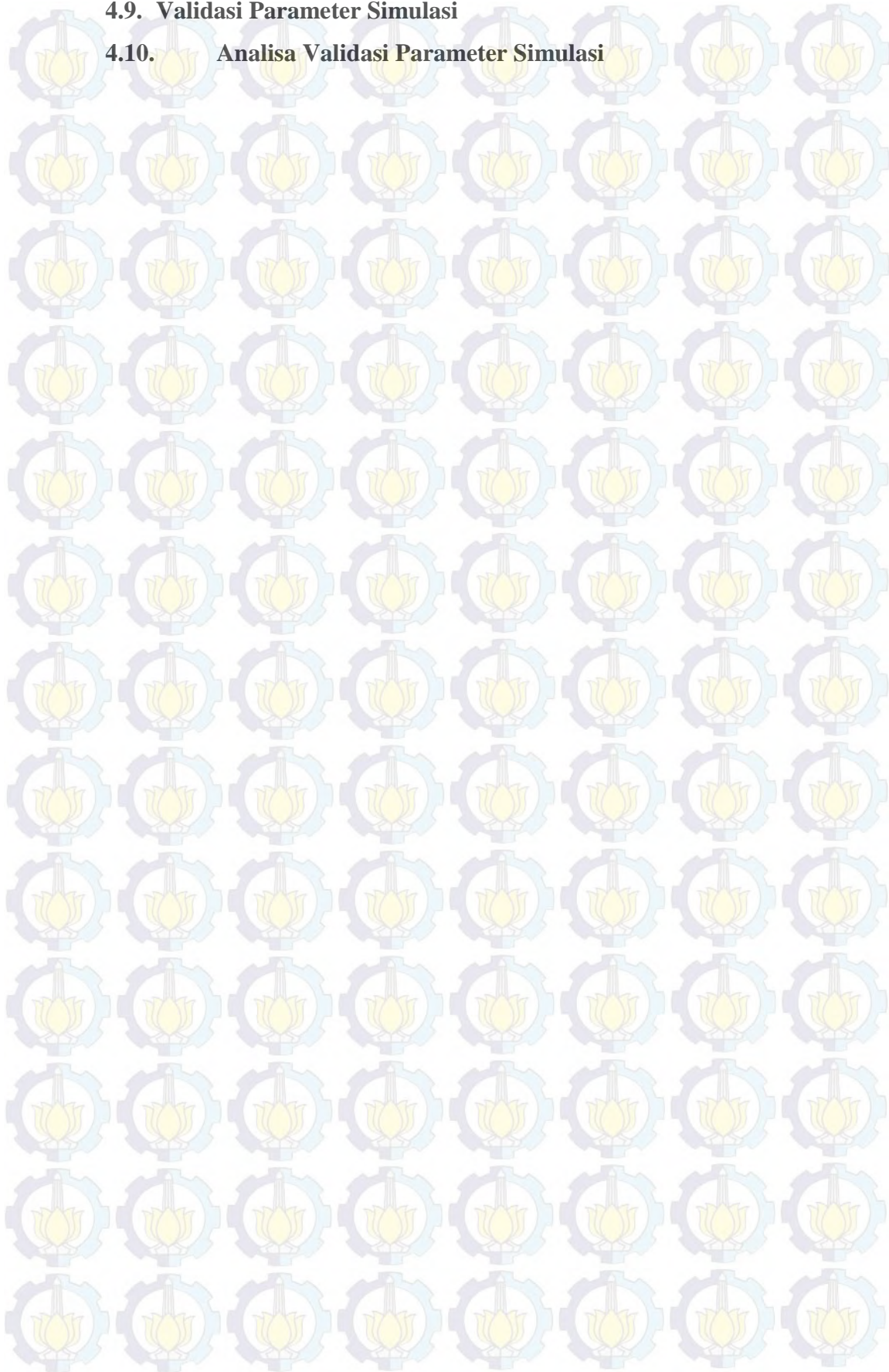
waktu 8 jam sedangkan pada kondisi dengan beban pada 30 menit pertama sisa kapasitas arus sebesar 194.3 mA sehingga rata-rata konsumsi arus tiap 30 menit akan berkurang sebanyak 26 mA maka baterai akan habis dalam waktu 4 jam.



(HALAMAN INI SENGAJA DIKOSONGKAN)

4.9. Validasi Parameter Simulasi

4.10. Analisa Validasi Parameter Simulasi



BAB 5 KESIMPULAN DAN SARAN

5.1. KESIMPULAN

Pada sub bab ini, akan dijelaskan mengenai kesimpulan-kesimpulan yang diambil berdasarkan hasil simulasi dan analisa dari bab 4 dengan metode bab 3. Kesimpulan yang dapat diambil adalah :

- 1) Dari hasil analisa proses manajemen energi pada sistem monitoring struktur jembatan dapat dilakukan dengan memilih topologi yang terbaik dengan routing yang baikpula.
- 2) Asumsi pertama adalah skenario dimana jarak antar node sebesar 100 meter x 12 meter. Didapatkan hasil sebagai berikut :
 - a. Topologi *star* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 139 packet *delay* sebesar 0.840 ms, *throughput* sebesar 0.65 kbps dan energi sebesar 0.106 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 142 packet, *delay* sebesar 0.532 ms, *throughput* sebesar 0,69 kbps dan energi sebesar 0.145 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *star* tidak dapat diimplementasikan pada sistem SMKS jembatan.
 - b. Topologi *mesh* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 2 packet, *delay* sebesar 27.25 ms, *throughput* sebesar 7.11 kbps dan energi sebesar 5.62 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 1 packet, *delay* sebesar 10.68 ms, *throughput* sebesar 21.28 kbps dan energi sebesar 5.99 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan.
 - c. Topologi *tree* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 2 packet, *delay*

sebesar 26.34 ms, *throughput* sebesar 6.35 kbps dan energi sebesar 5.19 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 9 *packet*, *delay* sebesar 19.50 ms, *throughput* sebesar 7.56 kbps dan energi sebesar 5.66 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan.

3) Asumsi kedua adalah skenario dimana jarak antar node sebesar 200 meter x 12 meter. Didapatkan hasil sebagai berikut :

- a. Topologi *star* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 152 *packet*, *delay* sebesar 0.940 ms, *throughput* sebesar 0.55 kbps dan energi sebesar 0.143 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 149 *packet*, *delay* sebesar 0.652 ms, *throughput* sebesar 0.48 kbps dan energi sebesar 0.199 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *star* tidak dapat diimplementasikan pada sistem SMKS jembatan.
- b. Topologi *mesh* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 4 *packet*, *delay* sebesar 27.45 ms, *throughput* sebesar 5.35 kbps dan energi sebesar 5.10 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 2 *packet*, *delay* sebesar 19.65 ms, *throughput* sebesar 19.89 kbps dan energi sebesar 5.31 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan.
- c. Topologi *tree* dengan routing DSDV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 5 *packet*, *delay* sebesar 25.97 ms, *throughput* sebesar 5.71 kbps dan energi sebesar 54.60 Joule. Sedangkan untuk routing AODV didapatkan nilai parameter kinerja jaringan sebagai berikut *packet loss* sebesar 8

packet, *delay* sebesar 20.43 ms, *throughput* sebesar 6.98 kbps dan energi sebesar 4.46 Joule. Berdasarkan parameter kinerja jaringan, maka topologi *mesh* dapat diimplementasikan pada sistem SMKS jembatan. Topologi yang dapat diterapkan pada SHM jembatan adalah topologi *mesh* dan *tree*.

- 4) Topologi yang terbaik dan dapat diterapkan pada Sistem SHM Jembatan pada penelitian ini adalah topologi *mesh*.
- 5) Pada asumsi pertama, penerapan topologi *mesh* dengan routing AODV lebih unggul dibandingkan topologi *star* dan *tree*.
- 6) Pada asumsi kedua, penerapan topologi *mesh* dengan routing AODV lebih unggul dibandingkan dengan topologi *star* dan *tree*.
- 7) Validasi yang dilakukan adalah validasi parameter simulasi dengan jarak 100 meter x 12 meter dan 200 meter x 12 meter sesuai dengan asumsi yang digunakan.

5.2. SARAN

Untuk kemajuan penelitian selanjutnya, terdapat beberapa saran setelah penelitian ini dilakukan. Saran yang dapat disampaikan adalah :

- 1) Untuk penelitian selanjutnya dapat menggunakan karakteristik sensor-sensor yang lain seperti sensor *strain gauge*, sensor suhu, dan *wind sensor*.
- 2) Pada penelitian ini hanya berbasis simulasi sehingga penelitian selanjutnya dapat menggunakan perangkat real dan dengan menguji topologi dan routing penelitian ini.
- 3) Untuk data real pada penelitian ini hanya sebagai validasi parameter dalam simulasi sehingga pada penelitian selanjutnya dapat menghitung data secara real sesuai hasil simulasi.



DAFTAR PUSTAKA

- [1] <http://www.tempo.co/read/news/2012/01/12/095376819/Ini-Versi-Resmi-Penyebab-Runtuhnya-Jembatan-Kutai>, diakses 20 Desember 2013.
- [2] S. Doebbling, C. Farrar, and M. Prime, “A summary Review of Vibration Based Damage Identification Methods”, *Shock and Vibration Digest*, vol. 30, pp. 91–105, 1998.
- [3] W. Dargie and C. Poellabauer, “Fundamentals of Wireless Sensor Networks: Theory and Practice”, John Wiley, 2010.
- [4] A. S. Kiremidjian, E. G. Straser, T. H. Meng, K. Law, and H. Soon, “Structural damage monitoring for civil structures”, in *Proc. Int. Workshop Struct. Health Monit.*, Stanford, CA, 1997, pp. 371–382.
- [5] Wei Chen, Miguel R. D. Rodrigues and Ian J. Wassell, “A Frechet Mean Approach for Compressive Sensing Date Acquisition and Reconstruction in Wireless Sensor Networks,” *IEEE Transactions on wireless communications*, vol. 11, no. 10, October 2012.
- [6] Akyldiz, I.F, Sankarasubramaniam, Y, dan Cayirci, E. (2002), “A Survey on Sensor Network”, *IEEE Commun Mag*, hal. 102-114.
- [7] Cardei Mihaela, My T. Thai, and Weili Wu, “Energy-Efficient Target Coverage in Wireless Sensor Networks”, *IEEE INFOCOM 2005*.
- [8] Y.K. Huang, A.C. Pang, P.C. Hsiu, W. Zhuang, and P. Liu, “Distributed Throughput Optimization for ZigBee Cluster-Tree Networks”, *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, 523-520, 2012.
- [9] Faried Achmad, “Evaluasi Performansi Protokol Routing Ad-Hoc On-Demand Distance Vector (AODV) Pada Jaringan Hybrid Ad-Hoc”, *Institut Teknologi Sepuluh November*, 2013.
- [10] Anjas Purnomo, Herman kurniawan, “Implementasi Protokol Routing Destination Sequenced Distance-Vector (DSDV) pada Jaringan Wireless NS-2 dengan NAM”, *Universitas Veteran Jawa Timur*. 2010.
- [11] S. K. Gupta, R. K. Saket, “Performance Metric Comparison of AODV and DSDV Routing Protocols in Manets Using NS-2”. Juni 2011.

- [12] C. Perkins, E. Belding-Royer and S. Das., "Ad-hoc On Demand Distance Vector(AODV) Routing". RFC 3561, IETF Network Working Group, July, 2003.
- [13] Septinurriandiani,"Sistem Monitoring Kesehatan Struktur-Penilaian Kondisi Dan Kriteria Peralatan Monitoring ",Kementerian Pekerjaan Umum (Pusat Penelitian Dan Pengembangan Jalan Dan Jembatan), 2011.
- [14] Bisby, L.A., ISIS Canada Educational Module No. 5: An Introduction to Structural Health Monitoring, ISIS Canada, www.isiscanada.com, 2006.
- [15] <http://telekom.ee.uui.ac.id/index.php/berita/15-wsn1>, diakses 20 Juni 2014.
- [16] <http://www.jennic.com/elearning/zigbee/files/html/module2/module2-3.htm>, diakses 20 Juni 2014.
- [17] E.N. Amalina, E. Setijadi dan Suwadi, "Perbandingan Topologi WSN (Wireless Sensor Network) Untuk Sistem Pemantauan Jembatan", Prosiding Conference on Smart-Green Technology in Electrical and Information Systems, Bali, 2013.
- [18] T.S.Fu, A. Ghosh, E. A. Johnson dan B. Krishnamachari, "Energy-efficient deployment strategies in structural health monitoring using wireless sensor networks", Structural Control and Health Monitoring, 2012.
- [19] M.J. Chae, H.S. Yoo, J.Y. Kim, M.Y. Cho, "Development of a WirelessSensor Network System for Suspension Bridge Health Monitoring", Elsevier Automation in Construction, vol.21, pp. 237-252, 2012.
- [20] D. Irawan dan R. Roestam, "Simulasi Model Jaringan Mobile Ad-Hoc (Manet) Dengan NS-3", Konferensi Nasional Sistem dan Informatika, Bali, 2011.
- [21] T. Liu and K. Liu, Improvement on DSDV in M obile Ad Hoc Networks, IEEE, China, 2007, pp. 1637-1640.
- [22] K.U.R Khan, A.V. Reddy, R.U. Zaman, K.A Reddy, T.S Harsha, An Efficient DSDV Routing Protocol for WirelessMobile Ad Hoc Networks and it's Performance Comparison, Second UKSIM European Symposium on Computer Modeling and Simulation, India, 2008, pp. 506-511.
- [23] A. H. Abd rahman dan Z. Ahmad, "Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad Hoc Networks", ISSN 1450-216X Vol.31 No.4, 2009, pp.566-576.

[24] http://www.libelium.com/es/smart_roads_wsn_smart_infrastructures/,
diunduh pada tanggal 20 Maret 2015.

[25] <http://www.decentlab.com/technology>, diunduh pada tanggal 20 Maret 2015.

[26] Amanaf, M. A., “Implementasi Sistem Akuisisi Data pada Bridge Structural Health Monitoring dengan Jaringan Sensor Nirkabel”, Institut Teknologi Sepuluh Nopember, 2014.

[27] Sikorsky, C., “Development of a Health Monitoring System for Civil Structures Using a Level IV Non-Destructive Damage Evaluation Method”, Proceedings of the 2nd International Workshop on Structural Health Monitoring 2000, Stanford University, CA, USA, pp. 68-81. 007, 1999.

[28] Faridatun N., Eko S. Dan Wirawan, “ Perbandingan Topologi dan Routing Untuk Memanajemen Konsumsi Daya Sensor pada Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan Berbasis WSN ”, Seminar Nasional : Sains, Rekayasa & Teknologi UPH – Tangerang, Hal. II-21, 2015.

[29] Faridatun N., Eko S. Dan Wirawan, “ Implementasi Komunikasi Data dengan Protokol Adhoc Berbasis WSN untuk SMKs Jembatan ”, Seminar Teknologi dan Rekayasa (SENTRA) 2015 UMM - Malang, Vol. 2 Hal. IV-35, 2015

[30] Wirawan, A.B., Indarto, E., “Mudah Membangun Sendiri dengan Network Simulator-2 (NS-2), Penerbit Andi, 2004.

LAMPIRAN A

```
#
=====
# Define options
#
=====
set val(chan) Channel/WirelessChannel ;# Channel Type
set val(prop) Propagation/TwoRayGround ;# radio-
propagation model
set val(netif) Phy/WirelessPhy/802_15_4
set val(mac) Mac/802_15_4
set val(ifq) Queue/DropTail/PriQueue ;# interface
queue type
set val(ll) LL ;# link layer
type
set val(ant) Antenna/OmniAntenna ;# antenna
model
set val(ifqlen) 150 ;# max packet
in ifq
set val(nn) 26 ;# number of
mobilenodes
set val(rp) AODV ;# routing
protocol
set val(x) 1500
set val(y) 50
# =====
# =====
# Antenna Settings
# =====
Antenna/OmniAntenna set X_ 0
Antenna/OmniAntenna set Y_ 0
Antenna/OmniAntenna set Z_ 0.6682
Antenna/OmniAntenna set Gt_ 2.1
Antenna/OmniAntenna set Gr_ 2.1
```

```

#####
#Physical layer setting
#####
Phy/WirelessPhy set freq_ 2.4e+9 ;# The working band is 2.4GHz
Phy/WirelessPhy set L_ 1.0 ;#Define the system loss in
TwoRayGround
Phy/WirelessPhy set pt_ 0.01
Phy/WirelessPhy set RXThresh_ 1.0e-13

# Initialize Global Variables
set ns_ [new Simulator]

# Tell teh simulator to use teh new type trace data
set tesis200 [open tesis200.tr w]
$ns_ trace-all $tesis200

#Define the NAM output file
set tesis200 [open tesis200.nam w]
$ns_ namtrace-all-wireless $tesis200 $val(x) $val(y)

# set up topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

# Create God
set god_ [create-god $val(nn)]
set chan_1_ [new $val(chan)]

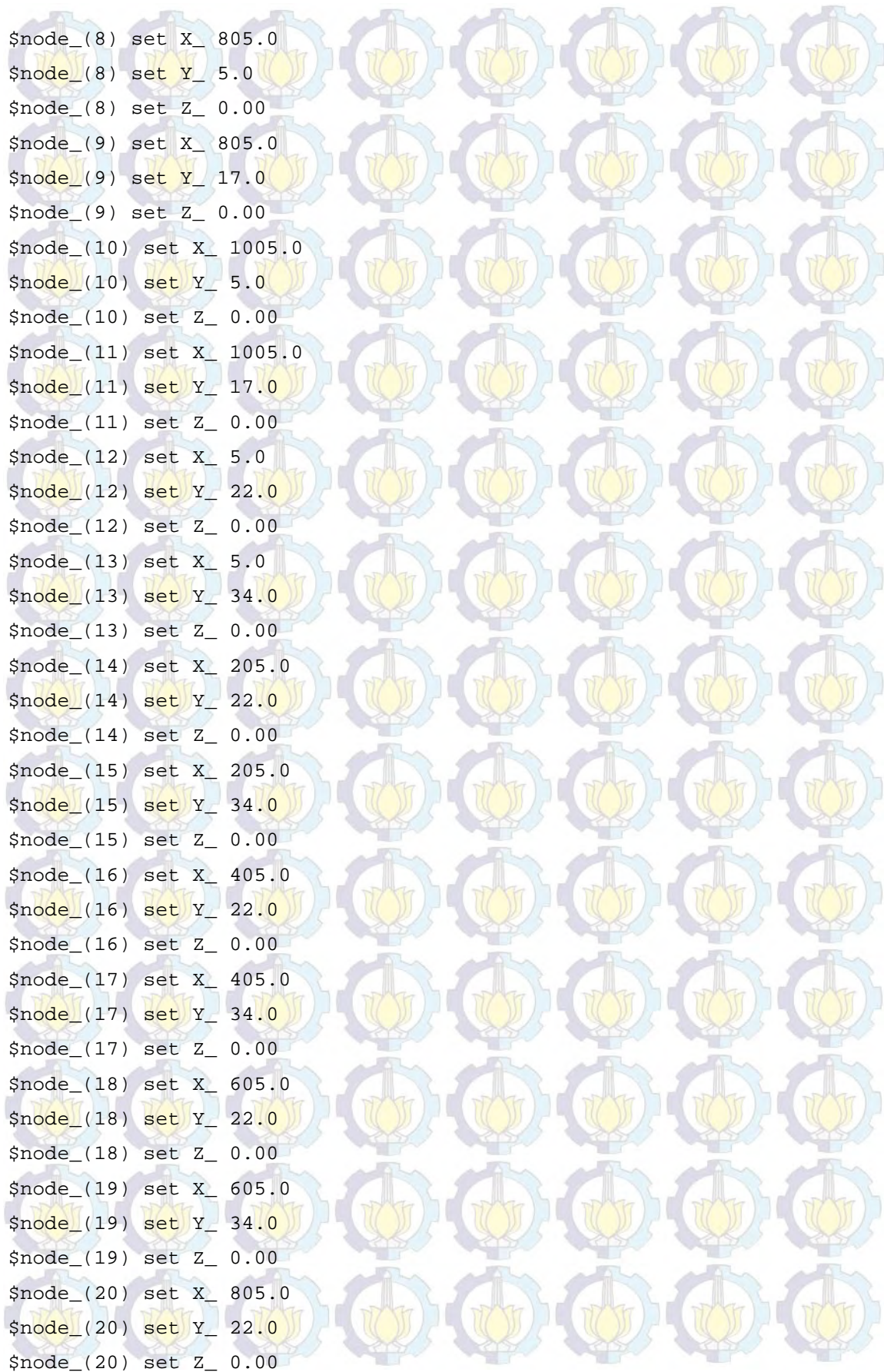
# configure node
$ns_ node-config -adhocRouting $val(rp) \
  -llType $val(ll) \
  -macType $val(mac) \
  -ifqType $val(ifq) \
  -ifqLen $val(ifqlen) \
  -antType $val(ant) \
  -propType $val(prop) \
  -phyType $val(netif) \

```

```

-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace OFF \
-energyModel "EnergyModel"\
-initialEnergy 19906\
-rxPower 0.875\
-txPower 0.18\
-channel $chan_1_
for {set i 0} {$i < $val(nn)} {incr i} {
set node_($i) [$ns_ node]
$node_($i) random-motion 0 ;
}
# Define the nodes positions
$node_(0) set X_ 5.0
$node_(0) set Y_ 5.0
$node_(0) set Z_ 0.00
$node_(1) set X_ 5.0
$node_(1) set Y_ 17.0
$node_(1) set Z_ 0.00
$node_(2) set X_ 205.0
$node_(2) set Y_ 5.0
$node_(2) set Z_ 0.00
$node_(3) set X_ 205.0
$node_(3) set Y_ 17.0
$node_(3) set Z_ 0.00
$node_(4) set X_ 405.0
$node_(4) set Y_ 5.0
$node_(4) set Z_ 0.00
$node_(5) set X_ 405.0
$node_(5) set Y_ 17.0
$node_(5) set Z_ 0.00
$node_(6) set X_ 605.0
$node_(6) set Y_ 5.0
$node_(6) set Z_ 0.00
$node_(7) set X_ 605.0
$node_(7) set Y_ 17.0
$node_(7) set Z_ 0.00

```



```
$node_(8) set X_ 805.0
$node_(8) set Y_ 5.0
$node_(8) set Z_ 0.00
$node_(9) set X_ 805.0
$node_(9) set Y_ 17.0
$node_(9) set Z_ 0.00
$node_(10) set X_ 1005.0
$node_(10) set Y_ 5.0
$node_(10) set Z_ 0.00
$node_(11) set X_ 1005.0
$node_(11) set Y_ 17.0
$node_(11) set Z_ 0.00
$node_(12) set X_ 5.0
$node_(12) set Y_ 22.0
$node_(12) set Z_ 0.00
$node_(13) set X_ 5.0
$node_(13) set Y_ 34.0
$node_(13) set Z_ 0.00
$node_(14) set X_ 205.0
$node_(14) set Y_ 22.0
$node_(14) set Z_ 0.00
$node_(15) set X_ 205.0
$node_(15) set Y_ 34.0
$node_(15) set Z_ 0.00
$node_(16) set X_ 405.0
$node_(16) set Y_ 22.0
$node_(16) set Z_ 0.00
$node_(17) set X_ 405.0
$node_(17) set Y_ 34.0
$node_(17) set Z_ 0.00
$node_(18) set X_ 605.0
$node_(18) set Y_ 22.0
$node_(18) set Z_ 0.00
$node_(19) set X_ 605.0
$node_(19) set Y_ 34.0
$node_(19) set Z_ 0.00
$node_(20) set X_ 805.0
$node_(20) set Y_ 22.0
$node_(20) set Z_ 0.00
```

```

$node_(21) set X_ 805.0
$node_(21) set Y_ 34.0
$node_(21) set Z_ 0.00
$node_(22) set X_ 1005.0
$node_(22) set Y_ 22.0
$node_(22) set Z_ 0.00
$node_(23) set X_ 1005.0
$node_(23) set Y_ 34.0
$node_(23) set Z_ 0.00
$node_(24) set X_ 255.0
$node_(24) set Y_ 20.0
$node_(24) set Z_ 0.00
$node_(25) set X_ 755.0
$node_(25) set Y_ 20.0
$node_(25) set Z_ 0.00
#-----
#Setup a UDP connection
set udp0 [new Agent/UDP]
$ns_ attach-agent $node_(0) $udp0
$udp0 set class_ 0
#Setup a CBR over UDP connection
set cbr0 [new Application/Traffic/CBR]
$cbr0 attach-agent $udp0
$cbr0 set packetSize_ 100
$cbr0 set interval_ 0.1
$cbr0 set rate_ 8000
set null0 [new Agent/Null]
$ns_ attach-agent $node_(25) $null0
$ns_ connect $udp0 $null0
#Schedule events for the CBR
$ns_ at 0.0 "$cbr0 start"
$ns_ at 360.0 "$cbr0 stop"
#Setup a UDP connection

```

```
set udp1 [new Agent/UDP]
$ns_ attach-agent $node_(1) $udp1
$udp1 set class_ 0

#Setup a CBR over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 100
$cbr1 set interval_ 0.1
$cbr1 set rate_ 8000

set null1 [new Agent/Null]
$ns_ attach-agent $node_(25) $null1
$ns_ connect $udp1 $null1

#Schedule events for the CBR
$ns_ at 360.0 "$cbr1 start"
$ns_ at 720.0 "$cbr1 stop"

#Setup a UDP connection
set udp2 [new Agent/UDP]
$ns_ attach-agent $node_(2) $udp2
$udp2 set class_ 0

#Setup a CBR over UDP connection
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp2
$cbr2 set packetSize_ 100
$cbr2 set interval_ 0.1
$cbr2 set rate_ 8000

set null2 [new Agent/Null]
$ns_ attach-agent $node_(25) $null2
$ns_ connect $udp2 $null2

#Schedule events for the CBR
$ns_ at 720.0 "$cbr2 start"
$ns_ at 1080.0 "$cbr2 stop"
```

```
#Setup a UDP connection
set udp3 [new Agent/UDP]
$ns_ attach-agent $node_(3) $udp3
$udp3 set class_ 0

#Setup a CBR over UDP connection
set cbr3 [new Application/Traffic/CBR]
$cbr3 attach-agent $udp3
$cbr3 set packetSize_ 100
$cbr3 set interval_ 0.1
$cbr3 set rate_ 8000

set null3 [new Agent/Null]
$ns_ attach-agent $node_(25) $null3
$ns_ connect $udp3 $null3

#Schedule events for the CBR
$ns_ at 1080.0 "$cbr3 start"
$ns_ at 1440.0 "$cbr3 stop"

#Setup a UDP connection
set udp4 [new Agent/UDP]
$ns_ attach-agent $node_(4) $udp4
$udp4 set class_ 0

#Setup a CBR over UDP connection
set cbr4 [new Application/Traffic/CBR]
$cbr4 attach-agent $udp4
$cbr4 set packetSize_ 100
$cbr4 set interval_ 0.1
$cbr4 set rate_ 8000

set null4 [new Agent/Null]
$ns_ attach-agent $node_(25) $null4
$ns_ connect $udp4 $null4

#Schedule events for the CBR
$ns_ at 1440.0 "$cbr4 start"
$ns_ at 1800.0 "$cbr4 stop"
```



```
#Setup a UDP connection
set udp5 [new Agent/UDP]
$ns_ attach-agent $node_(5) $udp5
$udp5 set class_ 0

#Setup a CBR over UDP connection
set cbr5 [new Application/Traffic/CBR]
$cbr5 attach-agent $udp5
$cbr5 set packetSize_ 100
$cbr5 set interval_ 0.1
$cbr5 set rate_ 8000

set null5 [new Agent/Null]
$ns_ attach-agent $node_(25) $null5
$ns_ connect $udp5 $null5

#Schedule events for the CBR
$ns_ at 1800.0 "$cbr5 start"
$ns_ at 2160.0 "$cbr5 stop"

#Setup a UDP connection
set udp6 [new Agent/UDP]
$ns_ attach-agent $node_(6) $udp6
$udp6 set class_ 0

#Setup a CBR over UDP connection
set cbr6 [new Application/Traffic/CBR]
$cbr6 attach-agent $udp6
$cbr6 set packetSize_ 100
$cbr6 set interval_ 0.1
$cbr6 set rate_ 8000

set null6 [new Agent/Null]
$ns_ attach-agent $node_(25) $null6
$ns_ connect $udp6 $null6

#Schedule events for the CBR
$ns_ at 2160.0 "$cbr6 start"
```

```
$ns_ at 2520.0 "$cbr6 stop"  
  
#Setup a UDP connection  
set udp7 [new Agent/UDP]  
$ns_ attach-agent $node_(7) $udp7  
$udp7 set class_ 0
```

```
#Setup a CBR over UDP connection  
set cbr7 [new Application/Traffic/CBR]  
$cbr7 attach-agent $udp7  
$cbr7 set packetSize_ 100  
$cbr7 set interval_ 0.1  
$cbr7 set rate_ 8000
```

```
set null7 [new Agent/Null]  
$ns_ attach-agent $node_(25) $null7  
$ns_ connect $udp7 $null7
```

```
#Schedule events for the CBR  
$ns_ at 2520.0 "$cbr7 start"  
$ns_ at 2880.0 "$cbr7 stop"
```

```
set udp8 [new Agent/UDP]  
$ns_ attach-agent $node_(8) $udp8  
$udp8 set class_ 0
```

```
#Setup a CBR over UDP connection  
set cbr8 [new Application/Traffic/CBR]  
$cbr8 attach-agent $udp8  
$cbr8 set packetSize_ 100  
$cbr8 set interval_ 0.1  
$cbr8 set rate_ 8000
```

```
set null8 [new Agent/Null]  
$ns_ attach-agent $node_(25) $null8  
$ns_ connect $udp8 $null8
```

```
#Schedule events for the CBR  
$ns_ at 2880.0 "$cbr8 start"
```

```
$ns_ at 3240.0 "$cbr8 stop"
set udp9 [new Agent/UDP]
$ns_ attach-agent $node_(9) $udp9
$udp9 set class_ 0

#Setup a CBR over UDP connection
set cbr9 [new Application/Traffic/CBR]
$cbr9 attach-agent $udp9
$cbr9 set packetSize_ 100
$cbr9 set interval_ 0.1
$cbr9 set rate_ 8000

set null9 [new Agent/Null]
$ns_ attach-agent $node_(25) $null9
$ns_ connect $udp9 $null9

#Schedule events for the CBR
$ns_ at 3240.0 "$cbr9 start"
$ns_ at 3600.0 "$cbr9 stop"

#Setup a UDP connection
set udp10 [new Agent/UDP]
$ns_ attach-agent $node_(10) $udp10
$udp10 set class_ 0

#Setup a CBR over UDP connection
set cbr10 [new Application/Traffic/CBR]
$cbr10 attach-agent $udp10
$cbr10 set packetSize_ 100
$cbr10 set interval_ 0.1
$cbr10 set rate_ 8000

set null10 [new Agent/Null]
$ns_ attach-agent $node_(25) $null10
$ns_ connect $udp10 $null10

#Schedule events for the CBR
$ns_ at 0.0 "$cbr10 start"
```

```

$ns_ at 360.0 "$cbr10 stop"
#Setup a UDP connection
set udp11 [new Agent/UDP]
$ns_ attach-agent $node_(11) $udp11
$udp11 set class_ 0
#Setup a CBR over UDP connection
set cbr11 [new Application/Traffic/CBR]
$cbr11 attach-agent $udp11
$cbr11 set packetSize_ 100
$cbr11 set interval_ 0.1
$cbr11 set rate_ 8000
set null11 [new Agent/Null]
$ns_ attach-agent $node_(25) $null11
$ns_ connect $udp11 $null11

#Schedule events for the CBR
$ns_ at 0.0 "$cbr11 start"
$ns_ at 360.0 "$cbr11 stop"

#-----
#Setup a UDP connection
set udp12 [new Agent/UDP]
$ns_ attach-agent $node_(22) $udp12
$udp12 set class_ 0
#Setup a CBR over UDP connection
set cbr12 [new Application/Traffic/CBR]
$cbr12 attach-agent $udp12
$cbr12 set packetSize_ 100
$cbr12 set interval_ 0.1
$cbr12 set rate_ 8000
set null12 [new Agent/Null]
$ns_ attach-agent $node_(24) $null12
$ns_ connect $udp12 $null12

```

```

#Schedule events for the CBR
$ns_ at 0.0 "$cbr12 start"
$ns_ at 360.0 "$cbr12 stop"

#Setup a UDP connection
set udp13 [new Agent/UDP]
$ns_ attach-agent $node_(23) $udp13
$udp13 set class_ 0

#Setup a CBR over UDP connection
set cbr13 [new Application/Traffic/CBR]
$cbr13 attach-agent $udp13
$cbr13 set packetSize_ 100
$cbr13 set interval_ 0.1
$cbr13 set rate_ 8000

set null13 [new Agent/Null]
$ns_ attach-agent $node_(24) $null13
$ns_ connect $udp13 $null13

#Schedule events for the CBR
$ns_ at 0.0 "$cbr13 start"
$ns_ at 360.0 "$cbr13 stop"

#Setup a UDP connection
set udp14 [new Agent/UDP]
$ns_ attach-agent $node_(20) $udp14
$udp14 set class_ 0

#Setup a CBR over UDP connection
set cbr14 [new Application/Traffic/CBR]
$cbr14 attach-agent $udp14
$cbr14 set packetSize_ 100
$cbr14 set interval_ 0.1
$cbr14 set rate_ 8000

set null14 [new Agent/Null]
$ns_ attach-agent $node_(24) $null14
$ns_ connect $udp14 $null14

```

```

#Schedule events for the CBR
$ns_ at 0.0 "$cbr14 start"
$ns_ at 360.0 "$cbr14 stop"

#Setup a UDP connection
set udp15 [new Agent/UDP]
$ns_ attach-agent $node_(21) $udp15
$udp15 set class_ 0

#Setup a CBR over UDP connection
set cbr15 [new Application/Traffic/CBR]
$cbr15 attach-agent $udp15
$cbr15 set packetSize_ 100
$cbr15 set interval_ 0.1
$cbr15 set rate_ 8000

set null15 [new Agent/Null]
$ns_ attach-agent $node_(24) $null15
$ns_ connect $udp15 $null15

#Schedule events for the CBR
$ns_ at 0.0 "$cbr15 start"
$ns_ at 360.0 "$cbr15 stop"

#Setup a UDP connection
set udp16 [new Agent/UDP]
$ns_ attach-agent $node_(18) $udp16
$udp16 set class_ 0

#Setup a CBR over UDP connection
set cbr16 [new Application/Traffic/CBR]
$cbr16 attach-agent $udp16
$cbr16 set packetSize_ 100
$cbr16 set interval_ 0.1
$cbr16 set rate_ 8000

set null16 [new Agent/Null]
$ns_ attach-agent $node_(24) $null16

```

```
$ns_ connect $udp16 $null16
#Schedule events for the CBR
$ns_ at 0.0 "$cbr16 start"
$ns_ at 360.0 "$cbr16 stop"

#Setup a UDP connection
set udp17 [new Agent/UDP]
$ns_ attach-agent $node_(19) $udp17
$udp17 set class_ 0

#Setup a CBR over UDP connection
set cbr17 [new Application/Traffic/CBR]
$cbr17 attach-agent $udp17
$cbr17 set packetSize_ 100
$cbr17 set interval_ 0.1
$cbr17 set rate_ 8000

set null17 [new Agent/Null]
$ns_ attach-agent $node_(24) $null17
$ns_ connect $udp17 $null17

#Schedule events for the CBR
$ns_ at 0.0 "$cbr17 start"
$ns_ at 360.0 "$cbr17 stop"

#Setup a UDP connection
set udp18 [new Agent/UDP]
$ns_ attach-agent $node_(16) $udp18
$udp18 set class_ 0

#Setup a CBR over UDP connection
set cbr18 [new Application/Traffic/CBR]
$cbr18 attach-agent $udp18
$cbr18 set packetSize_ 100
$cbr18 set interval_ 0.1
$cbr18 set rate_ 8000

set null18 [new Agent/Null]
```

```
$ns_ attach-agent $node_(24) $null18
$ns_ connect $udp18 $null18
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr18 start"
$ns_ at 360.0 "$cbr18 stop"
```

```
#Setup a UDP connection
set udp19 [new Agent/UDP]
$ns_ attach-agent $node_(17) $udp19
$udp19 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr19 [new Application/Traffic/CBR]
$cbr19 attach-agent $udp19
$cbr19 set packetSize_ 100
$cbr19 set interval_ 0.1
$cbr19 set rate_ 8000
```

```
set null19 [new Agent/Null]
$ns_ attach-agent $node_(24) $null19
$ns_ connect $udp19 $null19
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr19 start"
$ns_ at 360.0 "$cbr19 stop"
```

```
#Setup a UDP connection
set udp20 [new Agent/UDP]
$ns_ attach-agent $node_(14) $udp20
$udp20 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr20 [new Application/Traffic/CBR]
$cbr20 attach-agent $udp20
$cbr20 set packetSize_ 100
$cbr20 set interval_ 0.1
$cbr20 set rate_ 8000
```



```
set null20 [new Agent/Null]
$ns_ attach-agent $node_(24) $null20
$ns_ connect $udp20 $null20
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr20 start"
$ns_ at 360.0 "$cbr20 stop"
```

```
#Setup a UDP connection
set udp21 [new Agent/UDP]
$ns_ attach-agent $node_(15) $udp21
$udp21 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr21 [new Application/Traffic/CBR]
$cbr21 attach-agent $udp21
$cbr21 set packetSize_ 100
$cbr21 set interval_ 0.1
$cbr21 set rate_ 8000
```

```
set null21 [new Agent/Null]
$ns_ attach-agent $node_(24) $null21
$ns_ connect $udp21 $null21
```

```
#Schedule events for the CBR
$ns_ at 0.0 "$cbr21 start"
$ns_ at 360.0 "$cbr21 stop"
```

```
#Setup a UDP connection
set udp22 [new Agent/UDP]
$ns_ attach-agent $node_(12) $udp22
$udp22 set class_ 0
```

```
#Setup a CBR over UDP connection
set cbr22 [new Application/Traffic/CBR]
$cbr22 attach-agent $udp22
$cbr22 set packetSize_ 100
$cbr22 set interval_ 0.1
$cbr22 set rate_ 8000
```

```

set null122 [new Agent/Null]
$ns_ attach-agent $node_(24) $null122
$ns_ connect $udp22 $null122

#Schedule events for the CBR
$ns_ at 0.0 "$cbr22 start"
$ns_ at 360.0 "$cbr22 stop"

#Setup a UDP connection
set udp23 [new Agent/UDP]
$ns_ attach-agent $node_(13) $udp23
$udp23 set class_ 0

#Setup a CBR over UDP connection
set cbr23 [new Application/Traffic/CBR]
$cbr23 attach-agent $udp23
$cbr23 set packetSize_ 100
$cbr23 set interval_ 0.1
$cbr23 set rate_ 8000

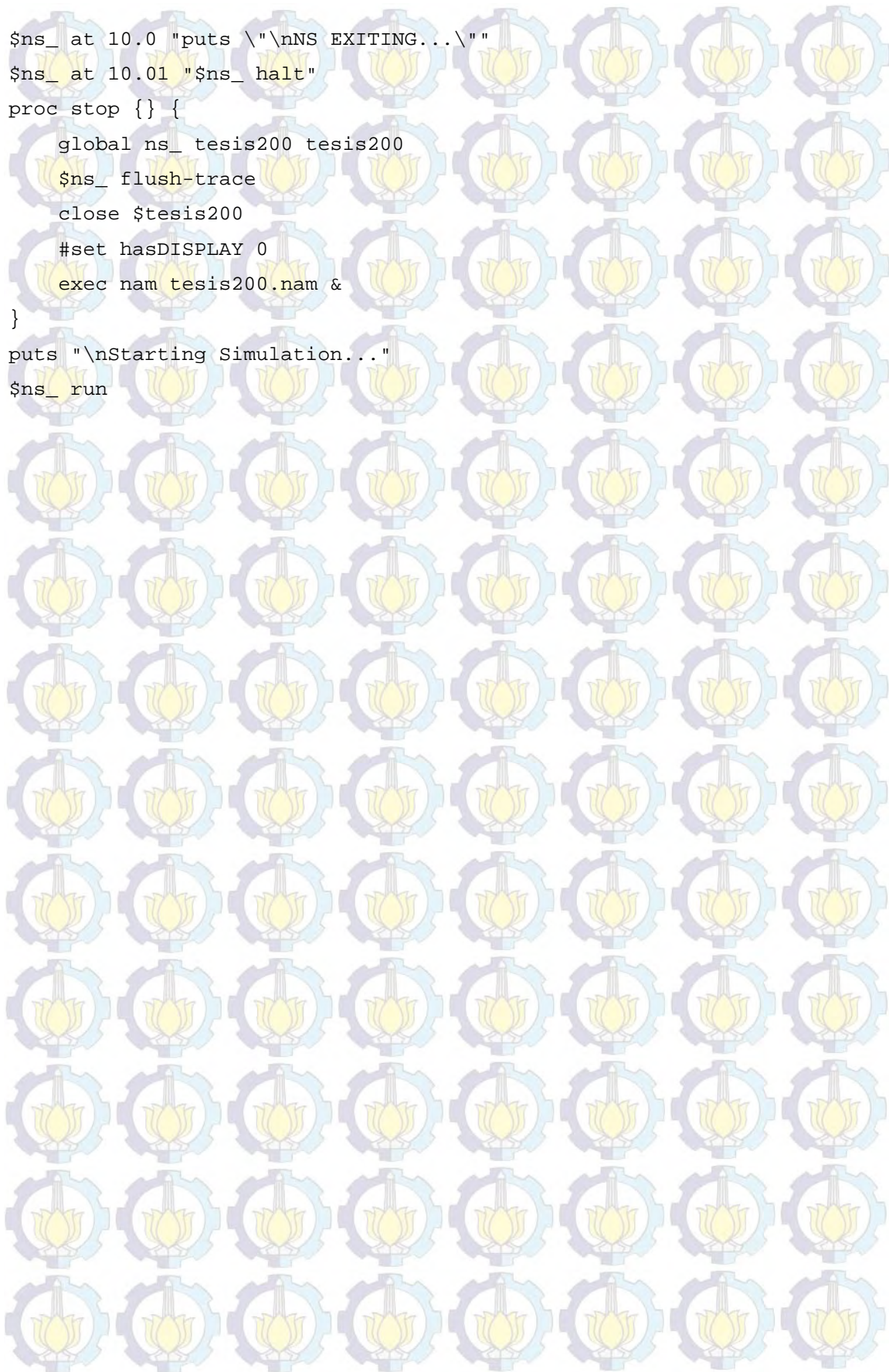
set null123 [new Agent/Null]
$ns_ attach-agent $node_(24) $null123
$ns_ connect $udp23 $null123

#Schedule events for the CBR
$ns_ at 0.0 "$cbr23 start"
$ns_ at 360.0 "$cbr23 stop"
#-----

# defines the node size in nam
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ initial_node_pos $node_($i) 100
}
# Tell nodes simulation ends at 60.0
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns_ at 60.0 "$node_($i) reset";
}
$ns_ at 10.0 "stop"

```

```
$ns_ at 10.0 "puts \"\nNS EXITING...\n\""  
$ns_ at 10.01 "$ns_ halt"  
proc stop {} {  
    global ns_ tesis200 tesis200  
    $ns_ flush-trace  
    close $tesis200  
    #set hasDISPLAY 0  
    exec nam tesis200.nam &  
}  
puts "\nStarting Simulation..."  
$ns_ run
```



LAMPIRAN B



Introduction

The MMA7361 from Freescale is a very nice sensor with easy analog interface. The MMA7361 is a 3.3V part and outputs an analog voltage for each of the three outputs. This voltage is in ratio to the measured acceleration and to the supply voltage (ratiometric). It has selectable sensitivity by dip switch. You will need some extra hardware to convert this analog signal to a usable digital one. The Arduino is really good option for it. This break board is especially designed for Arduino which has 3 JST connector that can be easily plug into our IO/Sensor expansion board.

Specification

- Voltage:3.3-8V
- Selectable sensitivity:±1. 5g/6g
- Low power:500 μ A @ measurement mode, 3 μ A @standby ;
- High sensivity: 800 mV/g @ 1.5g ;
- Interface:Analog Output
- Low pass filter
- Size:23x26mm
- Weight: 5 gram

LAMPIRAN C

Lampiran Program Koordinator :

```
void setup()
{
  Serial.begin(9600);
}

void loop()
{
  while (Serial.available() )
  {
    Serial.write(Serial.read()); // reply with whatever you receive
  }
}
```

Lampiran Program sensor:

```
#define TIME_HEADER "T"
#define TIME_REQUEST 7

int x2pin = 7;
int y2pin = 8;
int z2pin = 9;
float x2,y2,z2;

float vx, vy, vz, gx, gy, gz;
byte pesan[1];

byte b[30];
byte ACK[2];

int i;
int x,y;
```

```
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);}
void loop() {
  // put your main code here, to run repeatedly:
  Serial.print("@");
  Serial.print(" ");
  Serial.print("SENSOR1");
  Serial.print(" ");
  //digitalClockDisplay();
  //Serial.print(" ");
  dataSensor();
  Serial.println("#");
  delay(250); //<<<<<<<<<<<<< DELAY SAMPLING
}
void digitalClockDisplay () {
  Serial.print(hour());
  printDigits(minute());
  printDigits(second());
}
void printDigits(int digits){
  Serial.print(":");
  if(digits < 10)
    Serial.print('0');
  Serial.print(digits);
```

```

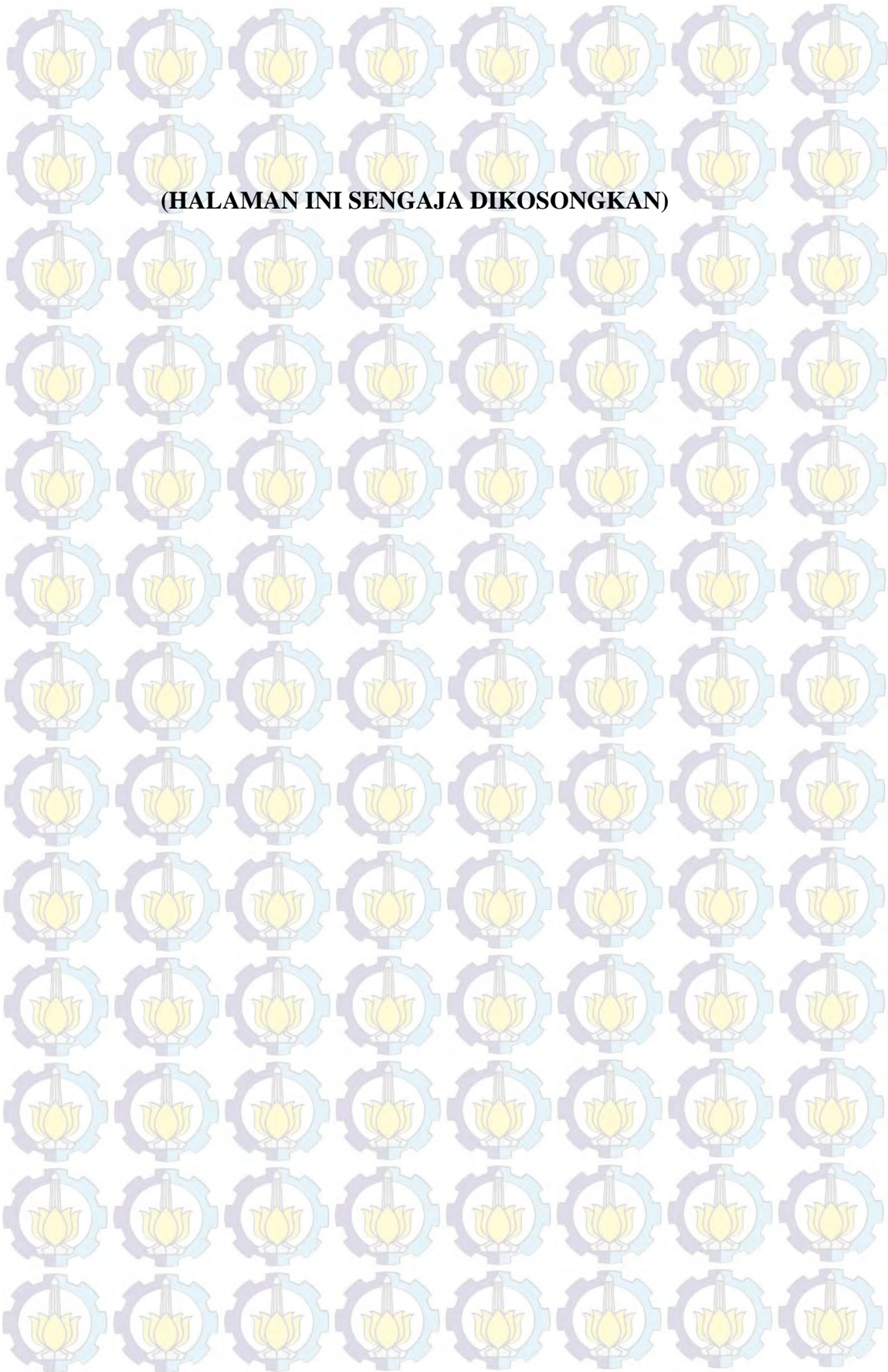
}
void dataSensor(){
  x2=analogRead(x2pin);
  y2=analogRead(y2pin);
  z2=analogRead(z2pin);
  analogReadResolution(16);
  //x2=((z/256)*5); //untuk 6g 0.206 1.5 g 0.8
  vx = (x2*5/65520)-2.50;
  gx = vx/0.206;
  vy = (y2*5/65520)-2.8;
  gy = vy/0.206;
  vz = (z2*5/65520)-3.3;
  gz = vz/0.206;
  digitalClockDisplay(); // jam
  Serial.print(" ");
  Serial.print(gx,4);
  Serial.print(" ");
  Serial.print(gy,4);
  Serial.print(" ");
  Serial.print(gz,4);
  Serial.print(" ");
}

```

Spesifikasi Modul RF XBee-PRO

Tabel 1 Spesifikasi Modul RF XBee-PRO

Performance	
Indoor Urban-Range	up to 300' (100 m)
Outdoor RF line-of-sight Range	up to 1 mile (1500 m)
Transmit Power Output (software selectable)	60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP
RF Data Rate	250,000 bps
Serial Interface Data Rate (software selectable)	1200 – 115200 bps (non-standard baud rates also supported)
Receiver Sensitivity	- 100 dBm (1% packet error rate)
Power Requirements	
Supply Voltage	2.8 – 3.4 V
Idle / Receive Current (typical)	55 mA (@3.3 V)
Power-down Current	< 10 μ A
General	
Operating Frequency	ISM 2.4 GHz
Frequency Band	2.4 - 2.4835 GHz
Modulation	OQPSK
Dimensions	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector
Networking & Security	
Supported Network Topologies	Point-to-point, Point-to multipoint & Peer-to-peer
Number of Channels	12 Direct Sequence Channels





BIODATA PENULIS



Penulis adalah anak kedua dari empat bersaudara yang lahir di Surabaya, 31 Desember 1991. Lulusan dari SMA Darul Ulum 2 RSBI Jombang, kemudian melanjutkan studinya di D4 Jurusan Teknik Telekomunikasi PENS-ITS di tahun 2009 hingga lulus pada tahun 2013.

Penulis melanjutkan pendidikan S2 di Teknik Elektro Bidang Studi Teknik Telekomunikasi Multimedia ITS karena mendapatkan beasiswa *fresh graduate* dari ITS pada tahun 2013. Di bawah bimbingan Bapak Eko Setijadi, ST., MT., Ph.D dan Bapak Dr. Ir. Wirawan, DEA., penulis mengambil tesis mengenai “*Desain Dan Analisa Manajemen Konsumsi Daya Pada WSN Untuk Sistem Monitoring Kesehatan Struktur (SMKS) Jembatan*”. Penulis dapat dihubungi melalui email faridatun.nadziroh@yahoo.com