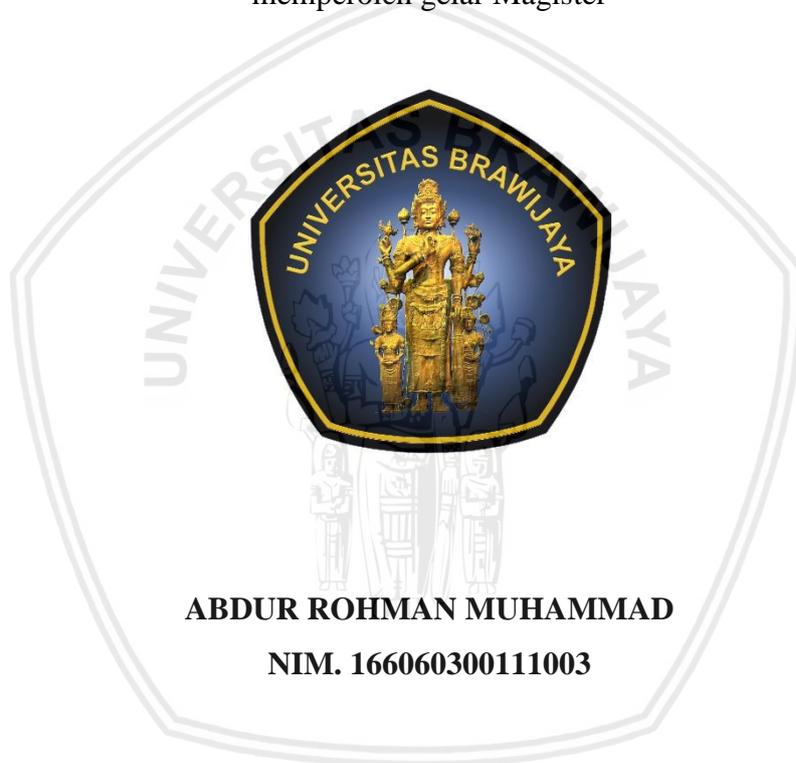


**EFISIENSI ENERGI WSN MENGGUNAKAN METODE
ASYNCHRONOUS DAN *ON-DEMAND DUTY-CYCLE***

**TESIS
TEKNIK ELEKTRO**

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Magister



**ABDUR ROHMAN MUHAMMAD
NIM. 166060300111003**

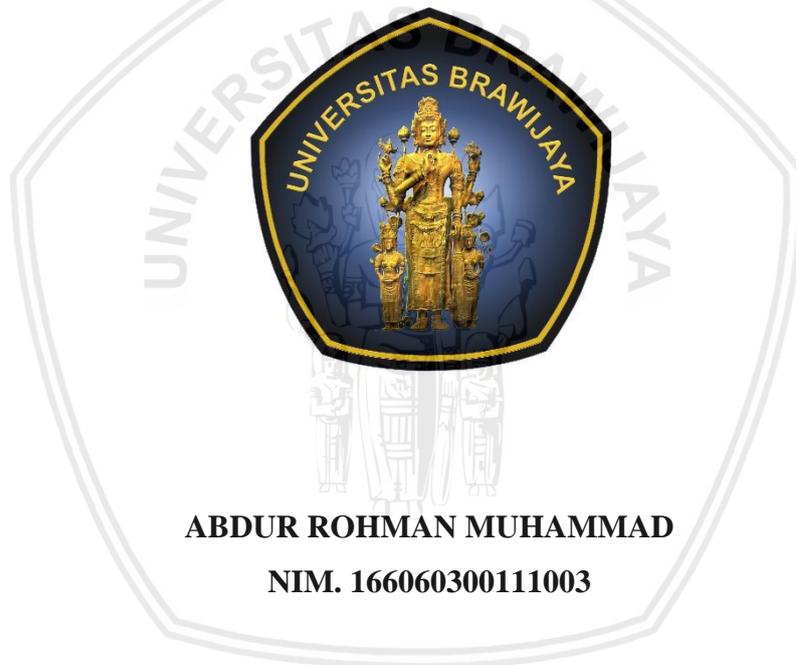
**UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
M A L A N G
2019**

**EFISIENSI ENERGI WSN MENGGUNAKAN METODE
ASYNCHRONOUS DAN ON-DEMAND DUTY-CYCLE**

TESIS

**PROGRAM MAGISTER TEKNIK ELEKTRO
MINAT SISTEM KOMUNIKASI DAN INFORMATIKA**

Ditujukan untuk memenuhi persyaratan
memperoleh gelar Magister



**ABDUR ROHMAN MUHAMMAD
NIM. 166060300111003**

**UNIVERSITAS BRAWIJAYA
FAKULTAS TEKNIK
M A L A N G
2019**

LEMBAR PENGESAHAN
PERNYATAAN ORISINALITAS



IDENTITAS PENGUJI TESIS

JUDUL TESIS :

EFISIENSI ENERGI WSN MENGGUNAKAN METODE *ASYNCHRONOUS* DAN *ON-DEMAND DUTY-CYCLE*

Nama Mahasiswa : Abdur Rohman Muhammad
NIM : 166060300111003
Program Studi : Magister Teknik Elektro
Minat (bila ada) : Sistem Komunikasi dan Informatika

KOMISI PEMBIMBING :

Ketua : Dr-Ing. Onny Setyawati, S.T., M.T., M.Sc.
Anggota : Achmad Basuki, S.T, M.MG, Ph.D

TIM DOSEN PENGUJI :

Dosen Penguji 1 : Dr. Ir. Bambang Siswojo, M.T.
Dosen Penguji 2 : Rahmadwati, S.T., M.T., Ph.D
Tanggal Ujian : 20 Desember 2019
SK Penguji : 2642 tahun 2019

RIWAYAT HIDUP

Penulis berasal dari Wajak, sebuah desa kecil 27 kilometer sebelah tenggara kota Malang. Lahir pada tanggal 24 November 1990. Setelah selesai menempuh pendidikan dasar di Kecamatan Wajak, penulis melanjutkan pendidikan menengah pertama dan atas di Kecamatan Turen. Selesai menamatkan sekolah, Penulis kemudian pindah ke Kota Malang untuk menempuh pendidikan tinggi di Jurusan Teknik Elektro Universitas Brawijaya. Selama kuliah, aktif dalam berbagai kegiatan di HMJ serta aktif menjadi asisten laboratorium Komputasi dan Jaringan hingga memperoleh gelar Sarjana Teknik pada tanggal 10 Agustus 2012.

Pada tahun 2011, sambil menyelesaikan kuliah, penulis bergabung dengan tim PPTI (tahun 2016 berubah menjadi Unit TIK), sebuah Unit Pelaksana Teknis di Universitas Brawijaya. Selama bekerja di Unit TIK, penulis mendapat banyak pengalaman pada bidang jaringan komputer, terutama yang berhubungan dengan *Routing*, *Wireless Access*, *AAA*, hingga *Network Service and Security*. Penulis juga mempunyai ketertarikan pada *Internet of Things* (IoT), hingga pada tahun 2016 penulis mendapat kesempatan izin belajar dari pimpinan untuk melanjutkan studi di jurusan yang sama.

Selama menempuh program Magister, penulis dipertemukan Allah dengan seorang perempuan, dan menikahinya pada bulan Agustus 2019, yang juga sedang berusaha mendapatkan gelar Master di Jurusan Kebidanan Fakultas Kedokteran. Penulis kini menetap di Kecamatan Lowokwaru Kota Malang.

Malang, 26 Desember 2019

Penulis

UCAPAN TERIMA KASIH

Dalam penyelesaian tesis ini penulis banyak mendapat bantuan dan perhatian yang tidak terhingga dari berbagai pihak. Untuk itu pada kesempatan ini penulis menyampaikan terima kasih kepada:

1. Allah ﷻ yang telah menggerakkan hati makhluk-makhluk-Nya sehingga segala urusan menjadi mudah.
2. Kedua orang tua dan seluruh anggota keluarga yang selalu mendukung dengan ikhlas.
3. Istriku, Bd. Sabatina Windyaningrum, yang selalu membantuku.
4. Dr-Ing. Onny Setyawati, S.T., M.T., M.Sc., sebagai pembimbing pertama yang telah banyak memberikan bimbingan dan kemudahan.
5. Achmad Basuki, S.T, M.MG, Ph.D., selaku pembimbing kedua dan pimpinan yang telah memberikan izin dan memberikan banyak ilmu serta pelajaran.
6. Raden Arief Setyawan, S.T., M.T. yang telah banyak membantu kelancaran kuliah saya.
7. Tim Recording TEUB, yang selalu membatu setiap kegiatan perkuliahan.
8. Rekan-rekan kerja saya yang memberikan ruang waktu kepada saya untuk dapat menyelesaikan kuliah.
9. Teman-teman kuliah program pascasarjana SKI-2016 yang selalu kompak.

Penulis menyadari meskipun penulis telah berusaha sebaik mungkin, masih banyak kekurangan dalam tesis ini. Jika ada kebaikan di dalam tulisan ini, kebaikan tersebut merupakan petunjuk dari Allah. Segala kekurangan dan kesalahan sungguh berasal dari kekhilafan penulis. Penulis berharap semoga tesis dapat bermanfaat bagi peneliti, bagi masyarakat Indonesia, serta bagi perkembangan dunia teknologi.

RINGKASAN

Abdur Rohman Muhammad, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya, Desember 2019, *Efisiensi Energi WSN Menggunakan Metode Asynchronous Dan On-Demand Duty-Cycle*, Dosen Pembimbing : Onny Setyawati dan Achmad Basuki.

Efisiensi energi pada *Wireless Sensor Network* (WSN) sangat dibutuhkan supaya *sensor-node* dapat bekerja dalam waktu lama dengan sumber energi yang terbatas. Penggunaan metode *duty-cycle* berdampak signifikan terhadap konsumsi energi pada sebuah *sensor-node*, dengan menjadwalkan waktu saat sebuah *sensor-node* harus aktif, *sleep*, mengirim, atau menerima data. Penjadwalan dapat dilakukan dengan metode *Synchronous*, *Asynchronous*, maupun *On-demand* yang menggunakan kanal tambahan.

Metode *on-demand* yang sebelumnya dilakukan, memiliki kelemahan pada radio pemicu (WuR) yang aktif terus-menerus meskipun tidak ada aktifitas pada jaringan. Penelitian ini menggabungkan penggunaan metode *On-demand* dengan melakukan penjadwalan pada WuR secara *Asynchronous*. Pada sisi penerima, mode *receive* pada WuR diaktifkan secara periodik, sedangkan pada sisi pengirim, sinyal *preamble* menggunakan WuR dengan informasi alamat tujuan disebar untuk mengaktifkan radio utama (MR) pada sisi penerima.

Hasil evaluasi pada 20 menit simulasi dengan penempatan *sensor-node* secara acak menunjukkan bahwa metode *Wake-up Asynchronous On-demand Radio Duty-cycle* (**waordc**) bekerja lebih efisien sebesar 3,61% (68,39 mWh) dari total energi yang dibutuhkan 1.890,49 mWh tanpa menggunakan penjadwalan, lebih baik dibandingkan metode **waco** sebesar 4,76% (90,15 mWh).

Kata kunci: WSN, penjadwalan, *duty-cycle*, *On-demand*, efisiensi energi.

SUMMARY

Abdur Rohman Muhammad, Electrical Engineering Department, Engineering Faculty, Universitas Brawijaya, December 2019, *WSN Energy Efficiency using Asynchronous and On-Demand Duty-Cycle Method*, Supervisor: Onny Setyawati and Achmad Basuki.

Energy efficiency on Wireless Sensor Networks (WSNs) is very important to make sensor-nodes work for a longer time with limited energy resources. The use of the duty-cycle method had a significant impact on energy consumption on a sensor-node, by scheduling when sensor-node works on active, sleep, transmit, or receive mode. Scheduling can be done by using the Synchronous method, Asynchronous, or On-demand using additional radio channels.

Previous works on the On-demand method had feebleness because they had always-on wake-up radio even there is no traffic among networks. This work combines uses of the On-demand method by doing another asynchronous scheduling for the wake-up radio. For the receiver side, receive-mode on wake-up radio is activated periodically. While on the sender side, by using WuR sent preamble signals. The signals are containing destination address information. Sensor-node with the matched address is awoken and sent interrupt to the main radio (MR).

Evaluation result for 20 minutes simulation with random placement shows that our *Wake-up Asynchronous On-demand Radio Duty-cycle (waordc)* method works more efficient 3,61% (68,39 mWh) compared to 1.890,49 mWh with no scheduling method, and better than the previous on-demand method, **Waco** 4,76% (90,15 mWh).

Keywords: WSN, scheduling, duty-cycle, On-demand, energy efficiency.

PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah ﷻ yang menguasai sekalian alam. Kami bersaksi bahwa tidak ada sesembahan yang patut disembah selain Allah yang Maha Esa yang tidak ada sekutu bagi-Nya dan kami bersaksi bahwa Muhammad ﷺ adalah hamba-Nya dan utusan-Nya. Semoga Allah ﷻ memberi sholawat dan salam kesejahteraan kepadanya, kepada keluarganya, dan para sahabatnya serta para pengikutnya.

Malang, 26 Desember 2019

Penulis



DAFTAR ISI

Lembar Pengesahan	iii
Pernyataan Orisinalitas	iii
Identitas Penguji Tesis	iv
Riwayat Hidup	v
Ucapan Terima Kasih	vi
Ringkasan.....	vii
Summary	viii
Pengantar.....	ix
Daftar isi.....	x
Daftar tabel.....	xiii
Daftar gambar	xiv
Daftar Simbol Dan Singkatan	xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Identifikasi Masalah	2
1.3. Rumusan Masalah.....	3
1.4. Batasan Masalah	3
1.5. Tujuan.....	3
1.6. Manfaat.....	3
BAB II DASAR TEORI	4
2.1. Penelitian Terkait.....	4
2.2. <i>Wireless Sensor Network (WSN)</i>	6
2.3. <i>Rime Protocol Stack</i>	7
2.4. Penjadwalan / <i>Duty-Cycle Scheduling</i>	7
2.4.1. <i>Synchronous Duty-cycle</i>	8
2.4.2. <i>Asynchronous Duty-cycle</i>	9
2.4.3. <i>On-Demand Duty-cycle</i>	11
2.5. <i>Ultra-low-Power Wake-Up Radio</i>	12
2.6. <i>CC2420 MOTES</i>	13
2.7. <i>ContikiOS</i>	13
2.8. <i>Cooja</i>	14
2.9. Penempatan dan Topologi jaringan sensor	15
2.10. Performa pada WSN	16



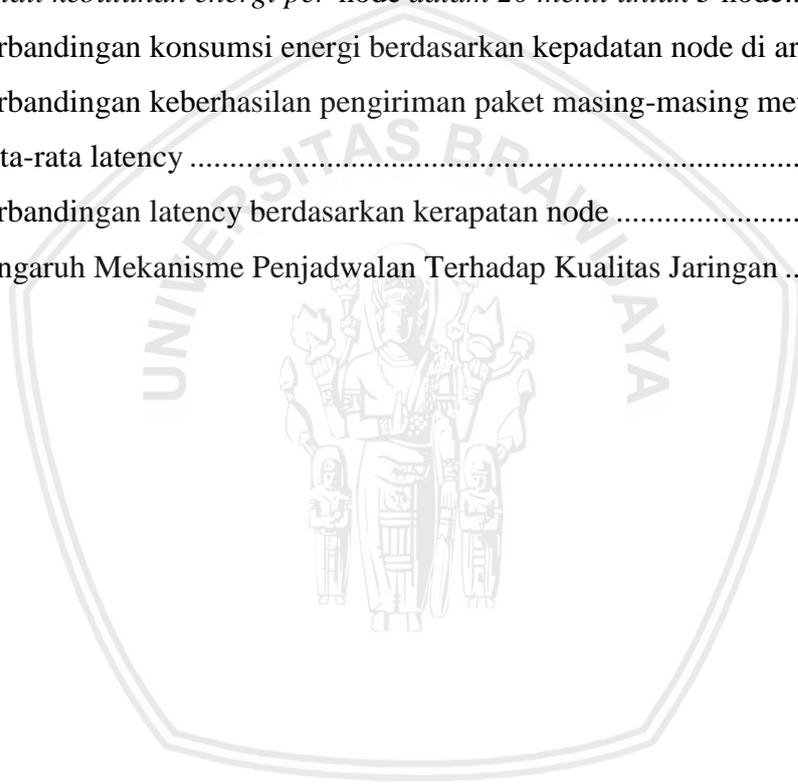
2.10.1.	Konsumsi Energi	16
2.10.2.	<i>Latency</i>	16
2.10.3.	Keberhasilan Pengiriman Paket.....	17
BAB III	KERANGKA KONSEP PENELITIAN	18
BAB IV	METODOLOGI PENELITIAN	21
4.1.	Studi Literatur.....	21
4.2.	Pemodelan Sistem.....	22
4.2.1.	Topologi Jaringan	22
4.2.2.	Identifikasi Penggunaan Energi.....	23
4.2.3.	Blok Diagram Komunikasi pada WSN	24
4.3.	Pembuatan Sistem.....	24
4.3.1.	<i>Protokol Stack</i>	25
4.3.2.	Komunikasi antar Node	25
4.3.3.	Pembuatan Modul waorcd	27
4.3.3.1.	PowerTracker untuk Radio Tambahan	27
4.3.3.2.	Mendaftarkan <i>protocol</i> Baru pada Netstack ContikiOS.....	28
4.3.3.3.	Menambahkan <i>Library</i> waorcd	28
4.3.3.4.	Menambahkan <i>Driver</i> waorcd	29
4.3.3.5.	Aplikasi <i>sensor_collect</i>	29
4.4.	Simulasi Sistem	30
4.4.1.	Spesifikasi Sistem.....	30
4.4.2.	Menjalankan Simulasi	30
4.5.	Pengujian	31
4.5.1.	Variabel Global.....	32
4.5.2.	<i>Dataset</i> Pengujian PAN.....	33
BAB V	HASIL DAN PEMBAHASAN	34
5.1.	Hasil Pengujian.....	34
5.2.	Pengujian Penggunaan Energi	35
5.2.1.	Detail Waktu Aktif <i>Node</i> dan Modul Radio.....	35
5.2.2.	Konsumsi Energi Keseluruhan <i>Node</i>	36
5.2.3.	Perbandingan Konsumsi Energi	37
5.3.	Pengujian Keberhasilan Pengiriman Paket (PDR)	38
5.4.	Pengujian <i>Latency</i>	39
5.5.	Pengaruh Mekanisme Penjadwalan Terhadap Kualitas Jaringan	41
BAB VI	KESIMPULAN DAN SARAN	43

6.1.	Kesimpulan.....	43
6.2.	Saran	43
DAFTAR PUSTAKA		44
LAMPIRAN.....		47



DAFTAR TABEL

Tabel 2.1 Waktu kerja node pada contikiOS	11
Tabel 2.2 Hasil pengukuran konsumsi energi yang dilakukan piyare dkk	13
Tabel 3.1 Perbandingan metode penjadwalan	20
Tabel 4.1 Penjabaran konsumsi energi pada WSN dengan mekanisme duty-cycle	23
Tabel 4.2 Parameter-parameter pengujian	32
Tabel 4.3 Hasil monitoring lingkungan untuk penanaman porang.....	33
Tabel 5.1 Hasil pengukuran waktu aktif selama pengujian 20 menit.	36
Tabel 5.2 <i>Detail kebutuhan energi per-node dalam 20 menit untuk 3 node</i>	37
Tabel 5.3 Perbandingan konsumsi energi berdasarkan kepadatan node di area 100x100m ²	38
Tabel 5.4 Perbandingan keberhasilan pengiriman paket masing-masing metode	38
Tabel 5.5 Rata-rata latency	40
Tabel 5.6 Perbandingan latency berdasarkan kerapatan node	41
Tabel 5.7 Pengaruh Mekanisme Penjadwalan Terhadap Kualitas Jaringan	42



DAFTAR GAMBAR

Gambar 2.1 Duty-cycle	4
Gambar 2.2 Metode asynchronous (a) El-Hoiydi (b) Buettner.....	5
Gambar 2.3 Pemanfaatan Wake-up Radio dengan daya rendah sebagai pemicu	6
Gambar 2.4 Kumpulan protokol yang menyusun Rime dan contoh komunikasi.....	7
Gambar 2.5 Contoh metode scheduled rendezvous	8
Gambar 2.6 Penggunaan preamble pada CX-MAC.....	9
Gambar 2.7 Waktu-waktu pada proses duty-cycle pada XMAC.....	10
Gambar 2.8 Sistem dual-radio dan garis waktu paket wake-up.....	12
Gambar 2.9 Desain penambahan ekstensi Wake-up Radio pada ContikiOS.....	12
Gambar 2.10 Protocol Stack ContikiOS	14
Gambar 2.11 Penempatan node dengan metode random and lattice.	15
Gambar 2.12 Latency antar node	16
Gambar 3.1 Kerangka penelitian.	18
Gambar 3.2 Klasifikasi teknik duty-cycle dan contohnya	19
Gambar 3.3 Hasil pengukuran waktu sleep antara cxmlac (kiri) dan waco (kanan).....	20
Gambar 4.1 Alur langkah penelitian	21
Gambar 4.2 Topologi tree dengan minimal sebuah relay-node	22
Gambar 4.3 Identifikasi energi dalam sebuah siklus duty-cycle pada setiap node.....	23
Gambar 4.4 Blok diagram komunikasi antar node pada WSN	24
Gambar 4.5 Sistem <i>preamble-checking</i> pada <i>wake-up radio</i>	24
Gambar 4.6 Activity-diagram modul asynchronous dan on-demand duty-cycle.	26
Gambar 5.1 Penempatan 3, 10, 25, dan 50 node.....	34
Gambar 5.2 Kebutuhan energi rata-rata sebuah sensor node selama 20 menit pengiriman .	37
Gambar 5.3 Grafik perbandingan keberhasilan pengiriman paket (PDR).....	39
Gambar 5.4 Latency pengiriman paket dengan kepadatan node rendah	40
Gambar 5.5 Perbandingan latency berdasarkan kepadatan jaringan	41



DAFTAR SIMBOL DAN SINGKATAN

Simbol/Singkatan	Keterangan
ContikiOS	sistem operasi jaringan yang dikembangkan Adam Dunkels
Cooja	Simulator untuk menjalankan
cxmac	Metode <i>asynchronous</i> yang dilakukan sebelumnya
E_{listen}	ketika keadaan menunggu paket
E_{mcu}	konsumsi energi oleh <i>microcontroller</i>
E_{rx}	ketika menerima paket
E_{sensor}	ketika melakukan pengambilan informasi
E_{sleep}	ketika dalam keadaan sleep
E_{switch}	pada saat proses awal radio
E_{tx}	ketika mengirim paket
IoT	<i>Internet of Things</i>
Motes	Sensor-sensor <i>node</i>
MR	Main radio / Radio Utama
M_{Rx}	Main radio pada mode <i>receive</i>
M_{Tx}	Main radio pada mode transmit
netstack	<i>Network stack</i> , kesatuan <i>protocol</i> jaringan
Node	Sebuah <i>endpoint</i> yang membentuk jaringan WSN
nullrdc	Modul null <i>duty-cycle</i> (mematikan <i>duty-cycle</i>)
P4I	Pusat Penelitian dan Pengembangan Porang Indonesia
PAN	Personal Area Network
PDR	<i>Packet Delivery Ratio</i> / Keberhasilan Pengiriman Paket
rime	Sebuah <i>protocol-stack</i> pada jaringan WSN
t_a	Jarak waktu setelah menerima paket <i>acknowledgment</i>
t_c	<i>Clear channel assessment</i>
t_d	Lama waktu untuk mendeteksi <i>acknowledgment</i> sedang dikirim
t_i	Interval setiap preamble
t_r	Waktu yang dibutuhkan untuk mendapatkan RSSI yang stabil
t_s	Waktu untuk mengirimkan paket terpendek
waco	Metode On-demand yang dilakukan sebelumnya
waordc	Modul <i>duty-cycle</i> yang diusulkan
WPAN	Wireless Personal Area Network
W_{Rx}	Wake-up radio pada mode <i>recieve</i>
WSN	Wireless Sensor Network
W_{Tx}	Wake-up radio pada mode transmit
WuR	Wake-up radio
wurrdc	Modul <i>duty-cycle</i> yang digunakan Waco



BAB I

PENDAHULUAN

1.1. Latar Belakang

Internet of Things (IoT) merupakan sebuah konsep yang memanfaatkan konektivitas internet dan *Personal Area Network (PAN)* untuk dapat melakukan pengamatan, kontrol, atau berbagi data pada benda-benda di dunia nyata. PAN dalam IoT dapat menggunakan komunikasi *wireless* karena kemudahannya dalam perancangan sistem komunikasi serta instalasi. Salah satu contoh penerapannya adalah *Wireless Sensor Network (WSN)*, jaringan sensor-sensor yang dapat saling berkomunikasi untuk bertukar data. Sebuah kesatuan sensor yang dapat berkomunikasi tersebut disebut *sensor-node* (Bagula, 2016).

Pada beberapa kondisi, *sensor-node* hanya dapat bekerja dengan sumber energi terbatas seperti baterai karena sulitnya akses untuk perawatan, minimnya cahaya matahari, atau angin sebagai sumber energi. Kondisi tersebut dialami pada implementasi sistem *monitoring* faktor lingkungan untuk penanaman Porang (*Amorphophallus muelleri*) yang dilakukan penulis bersama Pusat Penelitian dan Pengembangan Porang Indonesia (P4I) (Muhammad, 2018). Porang yang mempunyai kandungan *Glucomanan* tinggi ini mempunyai difersitas tinggi di wilayah Jawa Timur (Erta Puri Rosidiani, 2013). Tim P4I terus melakukan riset untuk dapat mengenal perilaku tanaman Porang dengan menanamnya di area Hutan Pendidikan Universitas Brawijaya yang terletak di lereng Gunung Arjuno Kecamatan Karangploso Kabupaten Malang. Diperlukan sebuah sistem untuk mendapatkan informasi tentang parameter lingkungan seperti suhu, kelembaban udara, kandungan air dalam tanah, dan intensitas cahaya sebagai dasar analisa perkembangan tanaman dan pengaruh lingkungan.

Supaya setiap *sensor node* pada *WSN* dapat bekerja dalam waktu lama, telah dilakukan penelitian untuk meningkatkan efisiensi penggunaan energi dari sisi perangkat keras maupun perangkat lunak. Pada sisi perangkat keras, dilakukan pemilihan baterai yang dapat bertahan lama, desain *micorcontroller*, desain antena, dan sistem transmisi yang sedikit mengkonsumsi arus listrik (Guru, Siddaraju, & Anandbabu, 2016)(Rault, Bouabdallah, & Challal, 2014). Metode lain dalam penghematan sumber energi tersebut adalah dengan melakukan penjadwalan waktu *sleep* pada *sensor-node*, sehingga *sensor-node* hanya bekerja ketika dibutuhkan saja. Penghematan dengan mekanisme *sleep/awake* atau lebih dikenal dengan mekanisme penjadwalan *duty-cycle* adalah salah satu metode yang dapat mengurangi konsumsi energi akibat *idle-state* dengan mengoperasikan radio pada *sensor-node* dalam mode *sleep*.

repository.ub.ac.id

Mekanisme penjadwalan secara umum dapat dikategorikan menjadi tiga yaitu *scheduled rendezvous*, *asynchronous*, dan *on-demand* (Chukwuka & Arshad, 2013). *Scheduled rendezvous* (Ye, Heidemann, & Estrin, 2002)(Tay, Jamieson, & Balakrishnan, 2004) lebih sederhana dengan melakukan sinkronisasi untuk menentukan “rentang waktu komunikasi” selain waktu tersebut seluruh node beroperasi dalam mode *sleep*. Namun *latency* meningkat seiring bertambahnya beban lalu-lintas. Pada metode *asynchronous* (El-Hoiydi, 2007)(Rajendran et al., 2009), node secara independen menentukan kapan waktu-komunikasi dengan *neighbor-node* melalui mekanisme *preamble* dan *probe*. Pengiriman paket secara terus menerus dapat mengonsumsi banyak sumber daya apabila *receiver* tidak segera memberikan balasan. Sedangkan waktu-komunikasi pada metode *on-demand* ditentukan oleh *sensor-node* yang ingin melakukan pengiriman paket, dengan cara membangunkan *sensor-node* tujuan yang berada di area transmisinya.

Metode *on-demand* dengan memanfaatkan radio tambahan (Schurgers, Tsiatsis, Ganeriwal, & Srivastava, 2002; Schurgers, Tsiatsis, & Srivastava, 2002; Xue & Vaidya, 2004), secara umum mempunyai keunggulan dibanding yang lain karena mempunyai *latency* rendah. Permasalahannya adalah adanya *idle listening* pada radio *Wake-up receive* (W_{Rx}) (Piyare, Murphy, Kiraly, Tosato, & Brunelli, 2017). W_{Rx} tetap aktif menunggu *beacon* (Schurgers, Tsiatsis, & Srivastava, 2002) atau *tone* (Xue & Vaidya, 2004) dari *node* yang akan berkomunikasi. *Sensor-node* harus mengaktifkan radio tambahan pada kanal lain secara terus-menerus menunggu *sensor-node* lain yang ingin berkomunikasi. Bagaimana melakukan penjadwalan pada radio tambahan tersebut, sehingga penggunaan energi dapat ditekan?

Penelitian ini menggabungkan dua metode sebelumnya untuk meningkatkan efisiensi energi WSN dengan melakukan penjadwalan *on-demand* seperti metode waco (Piyare, Istomin, & Murphy, 2017) dan menambahkan mekanisme *asynchronous* pada *wake-up radio* sehingga konsumsi energi pada *sensor-node* dapat lebih hemat dari metode-metode *duty-cycle* sebelumnya.

1.2. Identifikasi Masalah

Efisiensi energi pada WSN menggunakan mekanisme *on-demand duty-cycle* membutuhkan radio pemicu/*Wake-up Radio* dengan daya rendah sebagai penerima *wake-up signal*. Sedangkan W_{Rx} tetap mengonsumsi banyak energi karena terus-menerus aktif meskipun tidak terdapat aktivitas jaringan.

1.3. Rumusan Masalah

Berdasarkan uraian latar belakang, maka masalah yang menjadi fokus dalam penelitian ini adalah melakukan penghematan energi dengan menambahkan penjadwalan radio pemicu /*Wake-up Radio* (WuR) pada metode *on-demand*. Secara lebih detail, rumusan masalah dalam penelitian ini dapat diuraikan sebagai berikut:

1. Bagaimana melakukan penjadwalan radio pemicu (WuR) pada metode *on-demand duty-cycle* untuk meningkatkan efisiensi energi pada WSN?
2. Bagaimana pengaruh penjadwalan tersebut terhadap kualitas jaringan dengan parameter uji berupa *Packet Delivery Ratio* (PDR) dan *latency*?

1.4. Batasan Masalah

Berdasarkan uraian rumusan masalah, maka ruang lingkup dalam penelitian ini dibatasi sebagai berikut:

1. Diasumsikan tidak terjadi fluktuasi konsumsi energi akibat faktor lingkungan.
2. Penelitian difokuskan pada komunikasi WPAN antar *sensor-node*.
3. Pengujian dilakukan dengan simulasi menggunakan perangkat lunak COOJA for ContikiOS versi 3.0 dan Skymote sebagai *sensor-node*.
4. Aplikasi WSN pada pengamatan lingkungan, sehingga *sensor-node* dianggap statis.
5. Sebaran sensor menggunakan lokasi acak dengan topologi *tree*.

1.5. Tujuan

Tujuan penelitian ini adalah untuk mendapatkan efisiensi energi listrik menggunakan metode *on-demand duty-cycle* pada WSN, dengan melakukan mekanisme penjadwalan radio pemicu(WuR).

1.6. Manfaat

Penelitian ini diharapkan dapat bermanfaat dalam pengembangan mekanisme efisiensi energi WSN sehingga sensor dapat bertahan lebih lama tanpa banyak mengurangi parameter keberhasilan dan *latency* pengiriman data.

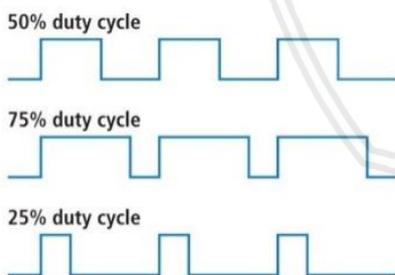
Selain itu, penelitian ini juga bermanfaat bagi pihak-pihak yang berkepentingan seperti Tim P4I (Pusat Penelitian dan Pengembangan Porang Indonesia) di kawasan UB Forest dalam menganalisis perubahan kondisi lingkungan penanaman Porang, serta mendapatkan rekam data yang bermanfaat dalam analisa lebih lanjut.

BAB II DASAR TEORI

2.1. Penelitian Terkait

Penelitian mengenai WSN sudah lama dilakukan, berbagai optimasi terus dikembangkan, penerapan pada pengamatan parameter lingkungan merupakan sub topik yang mendapat perhatian terutama di negara-negara berkembang seperti Indonesia. Studi mengenai *Precision Agriculture* yang telah dilakukan oleh Flores (Flores, Butaslac, Gonzales, Dumlao, & Reyes, 2017) dan Kassim (Kassim, Mat, & Harun, 2014) untuk mendapatkan efisiensi dalam produksi pertanian terbukti dapat menanggulangi pengaruh buruk lingkungan pada tanaman seperti pemberian pupuk berlebihan, atau mengetahui waktu panen yang tepat. Kodali dkk. (Kodali, Rawat, & Boppana, 2014) telah memilih jenis sensor serta melakukan pengujian terhadapnya.

Truong-Tien dkk. (Vo, Nguyen, & Vo, 2013) membagi menjadi 3 komponen utama dalam desain sistem *sensor-network* yang terdiri dari (1) beberapa *sensor-node* yang saling terhubung dalam sebuah *personal area networks (PAN)*, kemudian (2) *base-stations* yang menampung informasi dari *sensor-node* dan sekaligus menghubungkan dengan (3) pusat data dalam lingkup *wide-area networks (WAN)*. Kebutuhan energi pada *sensor-node* perlu diperhatikan karena sensor akan tersebar, sehingga perawatan seperti penggantian baterai dapat dilakukan seminimal mungkin.



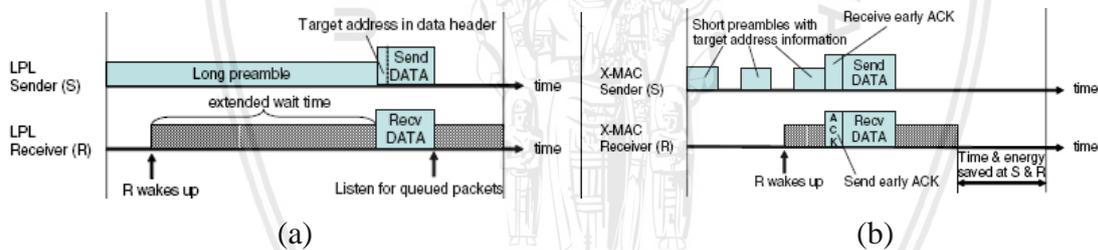
Gambar 2.1 Duty-cycle

Penghematan baterai dengan *duty-cycle* merupakan metode paling efektif dalam memperpanjang umur baterai. Seperti pada Gambar 2.1 dengan mengoperasikan *sensor* dalam kondisi *sleep* sehingga *sensor* melewati arus yang sangat kecil. Jaringan komunikasi *ad-hoc* tetap bisa terbentuk melalui *sensor-node* yang aktif secara bergantian. Jurdak dkk. (Jurdak, Ruzzelli, & O’Hare, 2010) menyebutkan bahwa total konsumsi energi yang dibutuhkan sebuah node adalah:

$$E = E_{mcu} + E_{listen} + E_{switch} + E_{tx} + E_{rx} + E_{sensor} + E_{sleep} \dots\dots\dots(1)$$

Pada persamaan 1 di atas, konsumsi energi dapat dibagi menjadi 7 kondisi; (1) E_{mcu} konsumsi energi oleh mikrokontroler, (2) E_{listen} ketika keadaan menunggu paket, (3) E_{switch} pada saat proses awal mulai radio, (4) E_{tx} ketika mengirim paket, (5) E_{rx} ketika menerima paket, (6) E_{sensor} ketika melakukan pengambilan informasi pada sensor, dan (7) E_{sleep} ketika dalam keadaan *sleep*.

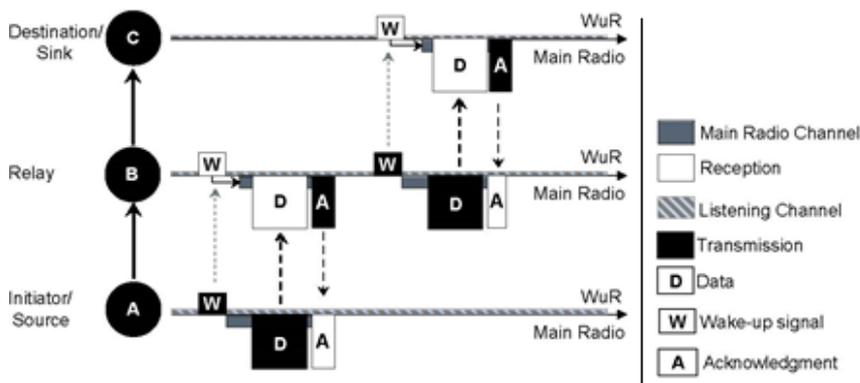
Teknik *duty-cycle* dapat diklasifikasikan menjadi tiga macam, *scheduled rendezvous*, *asynchronous*, dan *on-demand*. Metode *scheduled rendezvous* adalah cara paling sederhana di mana pengirim dan penerima memilih suatu waktu yang telah disepakati, pengirim akan mengirimkan informasi apabila memungkinkan, atau menunggu waktu berikutnya sehingga sinkronisasi waktu antar *node* menjadi sangat penting. Metode *asynchronous* memungkinkan setiap *node* menentukan waktu aktif secara independen tanpa perlu proses sinkronisasi waktu. Pengirim memancarkan sinyal tanda akan mengirimkan paket data, penerima secara berkala melakukan pengecekan apakah ada yang siap mengirimkan paket. Metode *on-demand* menggunakan *beacon* atau *tone* pada kanal khusus (bisa berupa radio tambahan) untuk mengaktifkan *neighbor* di sekitarnya sebelum mengaktifkan radio sesungguhnya yang akan mengirimkan paket.



Gambar 2.2 Metode *asynchronous* (a) El-Hoiydi (b) Buettner

Sumber: (El-Hoiydi, 2007)

El-Hoiydi (El-Hoiydi, 2007) mengusulkan metode *asynchronous* dengan mengirimkan sinyal *preamble* panjang. Buettner (Buettner, Yee, Anderson, & Han, 2006) mengganti sinyal *preamble* panjang tersebut dengan mengirim paket berisi informasi alamat target secara periodik seperti pada Gambar 2.2. Teknologi tersebut kemudian diadopsi sebagai salah satu modul *duty-cycle* dalam sistem operasi ContikiOS dengan nama CXMAC. Karena tujuan utama *preamble* hanya untuk memberitahu *receiver* bahwa *sender* akan melakukan pengiriman data, maka penggunaan radio sebagai *preamble* adalah mubazir.



Gambar 2.3 Pemanfaatan Wake-up Radio dengan daya rendah sebagai pemicu Sumber (Piyare, Istomin, et al., 2017)

Waco (*Wake-up Radio COOJA Extension*) (Piyare, Istomin, et al., 2017), merupakan metode *on-demand* yang diaplikasikan pada sistem operasi ContikiOS. Seperti pada Gambar 2.3, metode ini menggunakan dua radio di mana radio kedua (*Wake-up Radio*) membutuhkan daya yang lebih kecil. Radio tambahan tersebut secara pasif menunggu sinyal pemicu dari *node* lain yang ingin berkomunikasi. Namun, penggunaan *wake-up radio* yang terus menerus aktif dalam mode *receive* juga mengonsumsi energi dalam porsi yang besar apabila lalu lintas jaringan sangat sepi.

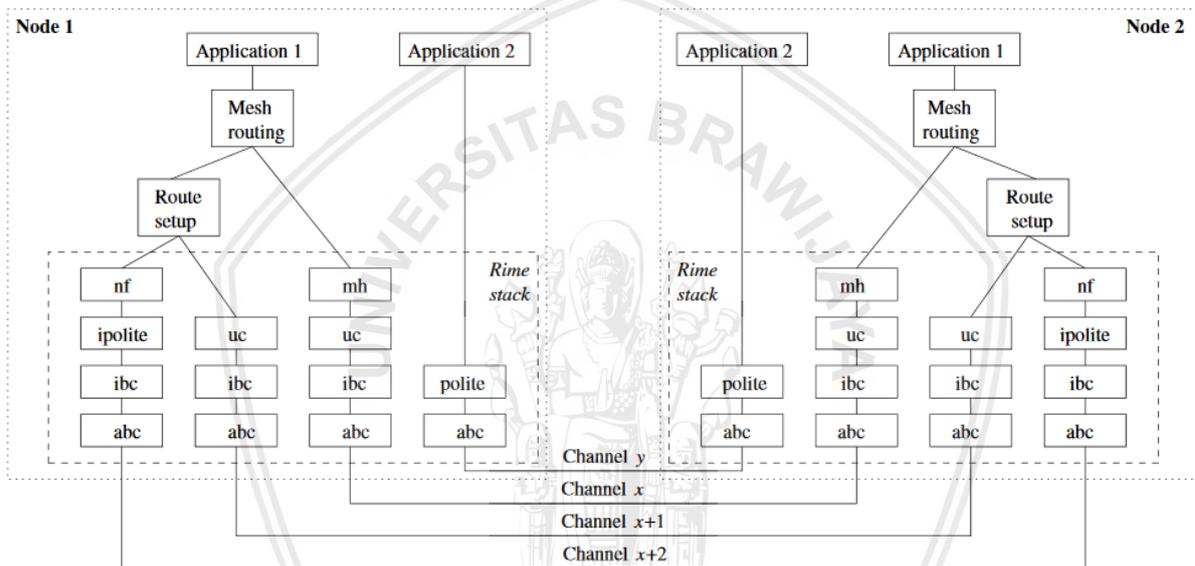
2.2. Wireless Sensor Network (WSN)

Wireless Sensor Network merupakan jaringan sensor yang mampu saling berkoordinasi sendiri. Sensor-sensor tersebut berukuran kecil dan biasa disebut *node* atau *mote*. *Node* terdiri dari sebuah mikrokontroler dengan kemampuan sangat sederhana, dengan prosesor dan memori terbatas. *Node* tersebut memiliki perangkat radio untuk berkomunikasi antar sesama *node*, sebuah sumber energi, dan satu atau beberapa sensor (Bagula et al., 2016). *WSN* merupakan jaringan komunikasi tanpa kabel yang terdiri dari beberapa sensor untuk memonitor atau melakukan kontrol terhadap kondisi fisis atau lingkungan tertentu. *WSN* terdiri dari beberapa *sensor-node* dan *base-station*. *Sensor-node* memiliki kemampuan untuk mengirimkan dan menerima data. Setiap *node* tersebar pada beberapa lokasi yang berbeda dan sulit untuk dijangkau (Mat, Kassim, & Harun, 2015). Jaringan *WSN* bersifat *ad-hoc* dan terdesentralisasi sehingga mempunyai skalabilitas tinggi dan kemudahan perawatan atau penggantian *node* yang rusak. *Base-station* berperan sebagai *sink-node*, pusat komunikasi dan sekaligus sebagai koordinator utama. *Base-station* juga berfungsi sebagai *gateway* yang menghubungkan dengan *Local Area Network* atau *Internet*.

2.3. Rime Protocol Stack

Rime merupakan beberapa kumpulan protokol yang menyediakan komunikasi yang ringan untuk komunikasi primitif (dengan kemampuan *hardware* terbatas) yang dapat bekerja mulai dari komunikasi *broadcast* dengan metode *best-effort* hingga jaringan dengan lalu lintas padat yang dapat diandalkan. Rime tersusun dalam beberapa lapisan protokol, di mana protokol yang lebih kompleks dibangun di atas protokol yang lebih sederhana.

Rime-stack mendukung komunikasi primitif baik *single-hop* maupun *multi-hop*. Rime tidak mengatur bagaimana pemilihan jalur pada *multi-hop*, akan tetapi memberitahu lapisan protokol di atasnya, atau lapisan aplikasi untuk menentukan hop berikutnya. Sehingga memungkinkan perubahan protokol *routing* di atas lapisan protokol *multi-hop* yang primitif.



Gambar 2.4 Kumpulan protokol yang menyusun Rime dan contoh komunikasi.

Sumber: (Dunkels, 2007)

Protokol atau aplikasi yang berjalan di atas Rime dapat menerapkan protokol yang tidak terdapat dalam kumpulan protokol Rime seperti pada Gambar 2.4. Apabila protokol atau aplikasi membutuhkan komunikasi primitif yang tidak ada pada Rime, maka sub protokol yang akan digunakan dapat diakses secara langsung..

2.4. Penjadwalan / Duty-Cycle Scheduling

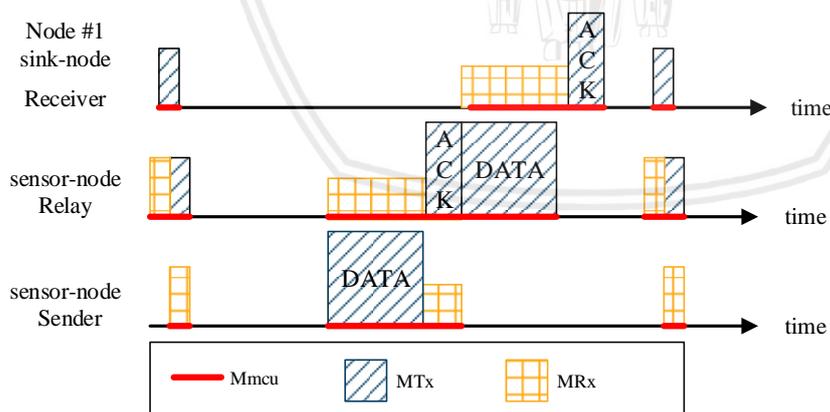
Efisiensi energi sangat penting pada WSN supaya umur jaringan bisa bertahan lama. Penghematan dari sisi *hardware* saja tidak cukup. Berdasarkan pengukuran, konsumsi energi pada *radio transceiver* menempati posisi tertinggi dibandingkan komponen lain di setiap *node* (Vo et al., 2013). Tmote Sky Board yang menggunakan modul *radio transeiver* CC2420 mengonsumsi sekitar 60 mW ketika dalam mode *listen*. Dengan menyalakan radio terus

menerus, baterai akan habis dalam beberapa hari saja. Untuk dapat menghemat konsumsi energi, radio harus lebih sering dalam kondisi mati. Namun, radio tersebut harus dalam kondisi aktif sebelum ada paket datang. Pengaturan waktu mati dan aktif dengan tetap menjaga keberhasilan komunikasi tersebut dikenal dengan teknik penjadwalan *duty-cycle*.

Tujuan utama penggunaan teknik *duty-cycle* adalah mematikan radio selama mungkin dengan menyediakan cukup waktu tertentu yang disepakati untuk berkomunikasi. Berikutnya akan diberikan penjelasan singkat mengenai metode-metode penentuan kesepakatan waktu tersebut, dengan metode *scheduled rendezvous*, *asynchronous* dan secara manual menyalakan radio dengan metode *on-demand*.

2.4.1. Synchronous Duty-cycle

Metode *synchronous* atau biasa disebut *scheduled rendezvous* adalah cara paling sederhana di mana pengirim dan penerima memilih suatu waktu yang telah disepakati, pengirim akan mengirimkan informasi apabila memungkinkan, atau menunggu waktu berikutnya. Ketepatan waktu sangat penting sehingga dibutuhkan mekanisme sinkronisasi waktu. Pada metode S-MAC (Ye et al., 2002) penerima secara periodik menyalakan radio, kemudian pengirim akan mengirimkan paket direntang periode tersebut. T-MAC (Wang & Liu, 2007) melakukan peningkatan waktu *sleep-state* dengan menambahkan mekanisme *time-out* apabila tidak terdapat paket masuk dalam jeda waktu tertentu (TA). Perbandingan kedua metode dasar tersebut diilustrasikan seperti pada Gambar 2.5 (Wang & Liu, 2007).



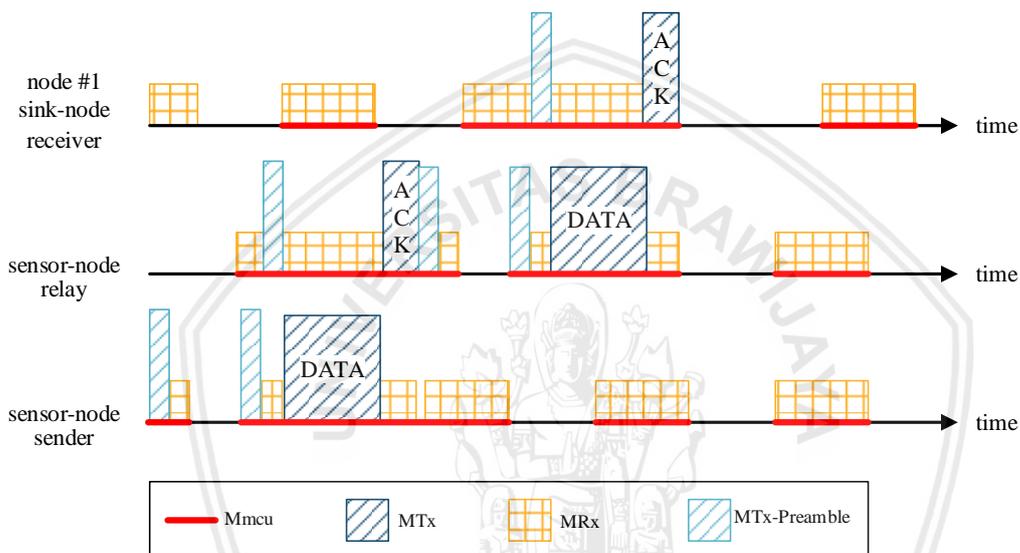
Gambar 2.5 Contoh metode *scheduled rendezvous*

Sumber: (Uwase, Bezunartea, Tiberghien, Dricot, & Steenhaut, 2017)

Penelitian yang menggunakan metode *scheduled rendezvous* antara lain adalah *Low-Power Listening/Probing* (LPL/LPP). Pada LPP *node* melakukan pengecekan apakah ada yang ingin berkomunikasi setiap t_1 detik. Apabila tidak terjadi komunikasi selama waktu pengecekan tersebut, *node* akan kembali dalam mode *sleep* untuk sampai t_1 detik berikutnya.

2.4.2. Asynchronous Duty-cycle

Pada metode *asynchronous, duty-cycle* dilakukan dengan cara menyalakan radio pada *receiver* secara periodik untuk mengetahui apakah ada aktivitas pada media komunikasi. Jika terdapat aktifitas, maka radio *receiver* akan tetap dalam kondisi aktif untuk menunggu pengiriman paket. Dari sisi pengirim, *sender* akan mengirimkan beberapa *strobe-packet* berisi informasi alamat penerima, dengan tujuan memberi informasi kepada *receiver* bahwa ada *neighbor* yang akan mengirimkan paket. Jarak antar *strobe* didefinisikan sepanjang jangka waktu *sleep* calon penerima, sehingga calon penerima tersebut akan aktif setelah menerima paket *strobe* dari *sender*. Tidak diperlukan adanya sinkronisasi waktu pada metode ini.



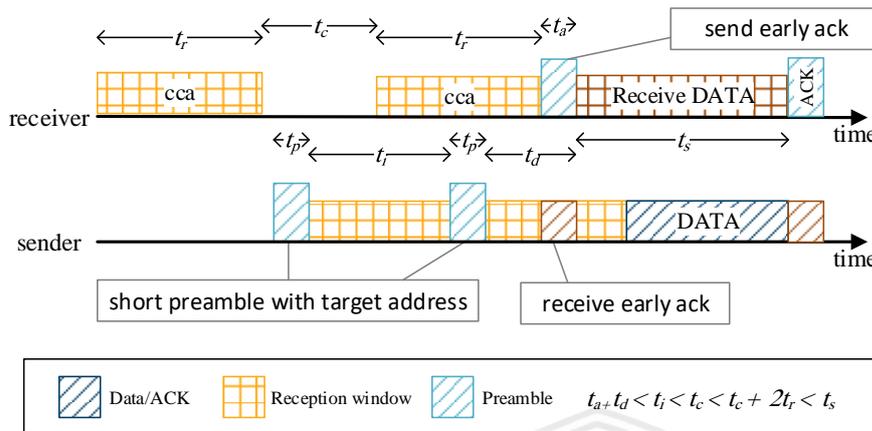
Gambar 2.6 Penggunaan preamble pada CX-MAC

Sumber: (Merlin & Heinzelman, 2010)

Pengecekan pada metode *asynchronous* bisa berupa *preamble-check* seperti pada metode B-MAC. Pengirim akan mengaktifkan radio receiver (Rx) menunggu *node* lain aktif untuk menerima paket. Metode ini mengonsumsi banyak energi karena *preamble-check* bisa jadi sangat lama. Permasalahan tersebut sudah diselesaikan dengan menggunakan pengulangan pengiriman data yang sama seperti pada metode CX-MAC Gambar 2.6. Jika terdapat paket lain, pengiriman berikutnya dilakukan setelah *acknowledgement* (Merlin & Heinzelman, 2010).

Pengaturan tempo pada mekanisme *duty-cycle* CXMAC diilustrasikan seperti pada Gambar 2.7. Gambar tersebut mengilustrasikan batasan waktu dimana CXMAC beroperasi. t_i (interval antar setiap pengecekan kanal) harus lebih kecil dari t_c (*clear channel assessment*). Hal tersebut untuk memastikan apabila ada paket yang datang, paket tersebut dapat terbaca

pada saat pengecekan pertama atau pada berikutnya. Dengan kata lain, $t_p + t_i + t_p$ (*preamble check*) harus lebih kecil dari waktu transmisi paket terpendek (t_s).



Gambar 2.7 Waktu-waktu pada proses *duty-cycle* pada XMAC.

Sumber: (Uwase et al., 2017)

Apabila transmisi paket terdeteksi, X-MAC akan menerima keseluruhan paket, kemudian mengirimkan *acknowledgment* di level *link-layer*. Gambar 2.7 menampilkan waktu yang dibutuhkan sampai paket *acknowledgment* terkirimkan, t_a , dan waktu yang dibutuhkan sampai paket *acknowledgment* terdeteksi, t_d , merupakan batas terendah dari interval waktu pengecekan t_c . Sehingga pembagian waktu dalam satu siklus X-MAC dapat dituliskan seperti persamaan 2 :

$$t_a + t_d < t_i < t_c < t_c + 2 t_r < t_s \dots \dots \dots (2)$$

Pada standar IEEE 802.15.4, beberapa variabel dapat diisi dengan konstanta sesuai yang telah didefinisikan di 802.15.4. Pertama, t_a telah didefinisikan sebagai 12 simbol panjang, setiap simbol membutuhkan waktu sebanyak $4/250$ ms, sehingga nilai t_a dapat diisi dengan $48/250$ atau $0,192$ ms. Berikutnya *receiver* pada IEEE 802.15.4 baru dapat mendeteksi penerimaan *acknowledgment* setelah *preamble* selama 4-byte dan 1-byte *frame* pemisah, atau $4*(8/250) + 1*(8/250)$ ms. Sehingga t_d dapat digantikan dengan nilai 0.16 ms. Nilai t_r juga disebutkan dalam *datasheet* CC2420 sebesar 0.192 ms. Dengan konstanta yang telah diketahui tersebut, maka persamaan 2 bisa dituliskan sebagai berikut :

$$0,352 < t_i < t_c < t_c + 0,384 < t_s \dots \dots \dots (3)$$

Variabel lain, t_i , t_c , dan t_s dapat ditentukan. Batas terendah pada persamaan 3 adalah $t_s > 0.736$ ms, untuk membatasi pada pengiriman paket terkecil yang dapat dikirimkan. Dengan menggunakan *bitrate* 250 kbps, maka setidaknya paket harus mempunyai ukuran 23 bytes, termasuk *preamble*, *frame* pembatas, dan panjang *field*, yang berukuran 16 bytes (Noreen, Bounceur, & Clavier, 2017). Implementasi pada ContikiOS menggunakan variabel berikut :

- $t_i = 0.4 \text{ ms}$,
- $t_c = 0.5 \text{ ms}$, dan
- $t_s = 0.884 \text{ ms}$ (28 - 6 bytes).

dalam satu siklus pengiriman data, sebuah node ditampilkan pada Tabel 2.1 berikut:

Tabel 2.1 Waktu kerja *node* pada contikiOS

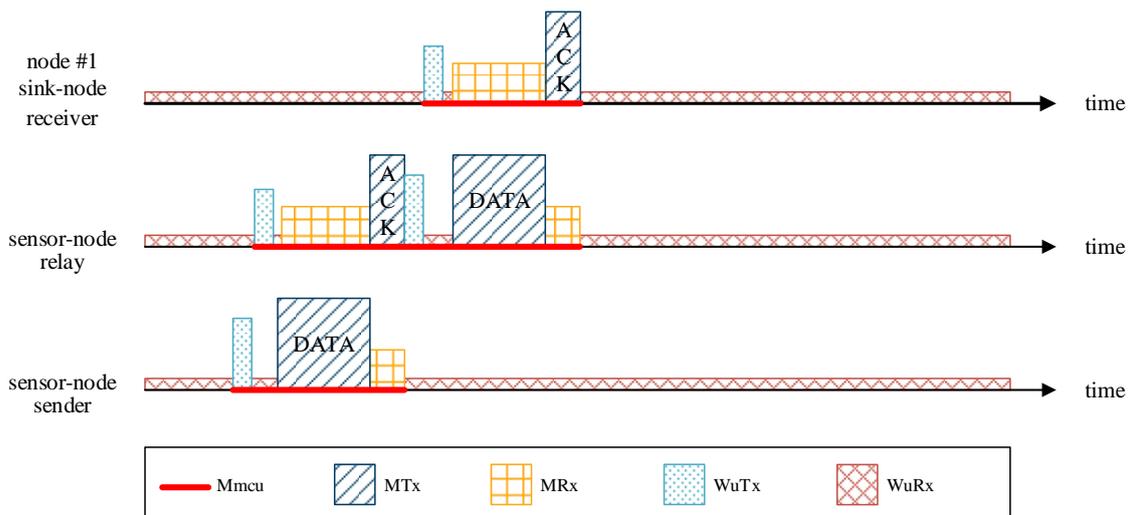
Var	Keterangan	Nilai
t_i	Interval setiap preamble	0.4 ms
t_r	Waktu yang dibutuhkan untuk mendapatkan RSSI yang stabil	0.192 ms
t_c	<i>Clear channel assessment</i>	0.5 ms
t_a	Jarak waktu setelah menerima paket dan mengirim <i>acknowledgment</i>	0.192 ms
t_d	Lama waktu untuk mendeteksi <i>acknowledgment</i> sedang dikirim	0.16 ms
t_s	Waktu untuk mengirimkan paket terpendek	0.884 ms

Protokol CX-MAC pada Contiki adalah pengembangan dari desain X-MAC. Dengan penambahan fitur *broadcast traffic*, *phase awareness*, *stream transport*, dan *optional MAC layer retransmissions*. X-MAC memperbaiki permasalahan *overhearing* yang disebabkan oleh panjangnya waktu pengiriman *preamble* ketika menunggu *receiver* siap mengirimkan data. X-MAC menggantinya dengan beberapa paket yang dikirimkan secara berkala berisi alamat tujuan.

Pada metode X-MAC, jika probabilitas paket diterima dalam satu rentang waktu telah diketahui, $P_{d(t)}$, maka parameter-parameter pada *sender* dan *receiver* dapat diatur dengan nilai optimal. Misalkan P_{Tx} , P_{Rx} , dan P_s mewakili daya yang dibutuhkan untuk mentransmisikan, menerima, dan *sleep*. S_p , S_{al} , dan S_d mewakili durasi waktu *sender* melakukan transmisi *preamble*, *acknowledgement listen*, dan *data* (isi dari paket). R_l and R_s mewakili lama waktu Penerima pada saat *listen* dan *sleep*.

2.4.3. On-Demand Duty-cycle

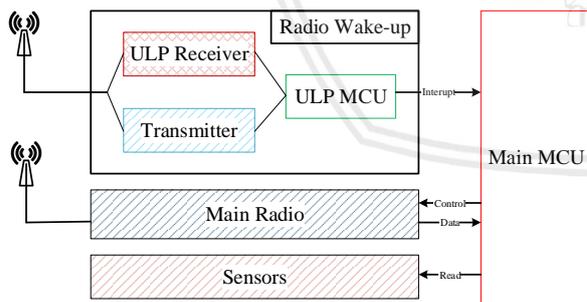
Metode *on-demand* mencoba mengurangi kerugian akibat proses *probing* maupun pengecekan secara periodik dengan menambahkan radio kedua sebagai pemicu seperti pada Gambar 2.8 Pemicu tersebut dapat berupa *beacon* (Schurgers, Tsiatsis, Ganeriwal, et al., 2002) atau *tone* (Schurgers, Tsiatsis, & Srivastava, 2002) pada kanal khusus berisi informasi sederhana sebatas alamat *neighbor* yang akan dituju. Pemicu tersebut dapat berupa radio tambahan atau *Wake-up Radio* (WuR). *Receiver* pada WuR bekerja dengan daya yang lebih rendah dibandingkan *receiver* radio utama. *Wake-up Radio* tersebut mengaktifkan *neighbors* spesifik di sebelahnya untuk mengaktifkan radio utama yang dapat mengirimkan paket yang lebih kompleks seperti data dan informasi routing.



Gambar 2.8 Sistem *dual-radio* dan garis waktu paket *wake-up*.
 Sumber: (Xue & Vaidya, 2004)

2.5. Ultra-low-Power Wake-Up Radio

Seiring berkembangnya teknologi perangkat keras, desain radio pemicu WuR juga mengalami optimasi dalam mengurangi konsumsi daya. Seperti pada Gambar 2.9, desain WuR lebih sederhana dibandingkan radio utama. WuR menggunakan *microcontroller* hemat energi untuk mengolah modulasi OOK. Ansari et al. (Ansari, Pankin, & Mähönen, 2009) menampilkan hasil pengukuran penggunaan arus ketika *sleep* sebesar $4.176 \mu\text{A}$ pada rangkaian modul *wake-up* yang terhubung pada *node* TelosB menggunakan unit *microcontroller* MSP430 untuk proses *address decoding*.



Gambar 2.9 Desain penambahan ekstensi *Wake-up Radio* pada ContikiOS
 Sumber: (Piyare, Istomin, et al., 2017).

Piyare dkk. telah menerapkan metode **waco** yang diusulkannya pada produk chip dari Texas Instrument CC2420 sebagai modul komunikasi utama berjalan pada frekuensi 2.4GHz dan menambahkan *transceiver* CC1101 sebagai radio tambahan yang berjalan di frekuensi Sub-1GHz. Hasil perbandingan konsumsi energi antara kedua modul tersebut ditampilkan pada Tabel 2.2:

Tabel 2.2 Hasil pengukuran konsumsi energi yang dilakukan piyare dkk

Parameter	Konsumsi Daya (mW)		
	Rx	Tx	Idle
CC2420	56,4	52,2	56,4
MSP430	1,44	8,4	1,944

Tabel 2.2 merupakan hasil pengukuran konsumsi energi di laboratorium menggunakan dua modul W_{Rx} dan W_{Tx} bekerja pada frekuensi ISM 868 MHz dengan sensitifitas penerima sebesar -55dBm dengan jarak komunikasi maksimum 50m. Daya maksimum yang dibutuhkan W_{Rx} adalah 1,44 mW ketika mode *receiving*. Sebesar 1,944 mW ketika *idle*, menunggu sinyal *wake-up*. Konsumsi daya W_{Tx} adalah 8.3 mW ketika mentransmisikan sinyal sebesar +10dBm.

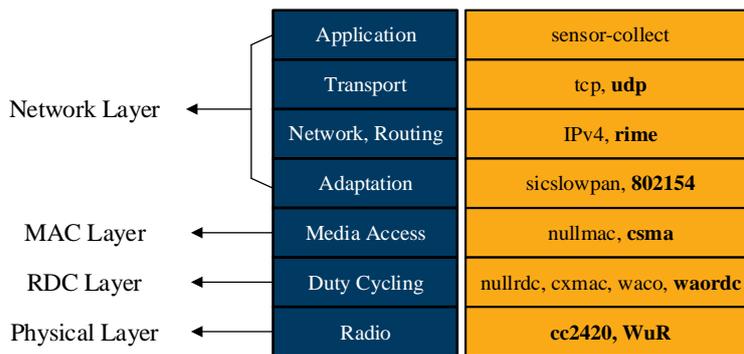
2.6. CC2420 MOTES

CC2420 *motes* adalah sebuah *single-chip* dengan radio transiver yang berjalan pada frekuensi 2.4 GHz sesuai standar IEEE 802.15.4. CC2420 *transceiver* didesain untuk penggunaan aplikasi *wireless* berdaya dan tegangan rendah dengan spesifikasi :

- Radio *transceiver Single-chip* 2.4 GHz menggunakan standar IEEE 802.15.4, mendukung MAC protokol, menggunakan modem baseband, 2 MChips/s dan 250 kbps *data rate*.
- Konsumsi daya rendah (RX: 18.8 mA, TX: 17.4 mA)
- Membutuhkan tegangan rendah (2.1 – 3.6 V) jika menggunakan internal voltage regulator, dan 1.6 – 2.0 V tanpa regulator.

2.7. ContikiOS

ContikiOS adalah sistem operasi jaringan yang dapat berjalan pada berbagai platform sensor yang bekerja dengan daya listrik rendah. ContikiOS menyediakan 3 macam *network stack* Ipv4, Ipv6, dan Rime. *Network stack* pada *ContikiOS* sedikit berbeda dengan model *4-layers* yang biasa diadopsi pada TCP/IP. Di antara lapisan antara *Physical* dan *Network layers*, terdapat 1 *layer* tambahan *Radio Duty-Cycle* (RDC). Gambar 2.10 berikut menunjukkan bagaimana pembagian *layer* pada ContikiOS.



Gambar 2.10 Protocol Stack ContikiOS

Sumber: (Cotrim & Kleinschmidt, 2017)

Untuk mendeteksi adanya *collision*, ContikiOS menggunakan mekanisme CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*). Membaca media jaringan sebelum mengirim, kemudian membatalkan apabila ada *node* lain yang sedang mengirim. Waktu pengiriman ulang tergantung pada RDC layer.

Radio Duty-Cycle (RDC) melakukan penghematan energi dengan mematikan radio transmisi dalam keadaan *sleep*. Lapisan ini sangat penting karena selain bertanggung jawab menentukan kapan paket akan dikirimkan, RDC juga bertanggung-jawab memastikan bahwa *node* harus dalam keadaan aktif ketika paket datang. Berikut ini adalah pembagian lapisan pada Sistem operasi ContikiOS beserta contoh *driver* yang memuat protokolnya:

1. *Network layer* - rime_driver
2. *MAC layer* - nullmac_driver
3. RDC - nullrdc_driver
4. *Framer* - framer_nullmac
5. *Radio* - nullradio_driver

2.8. Cooja

Cooja adalah perangkat lunak untuk melakukan simulasi jaringan ContikiOS. *Motes* dapat disimulasikan dalam lingkungan kerja Java Native Interfaces, meskipun lebih lambat tetapi mampu melakukan inspeksi secara detail dan presisi terhadap perilaku sistem. Cooja juga mampu melakukan simulasi *motes* dalam skala besar. Cooja memiliki tampilan utama yang menampilkan beberapa jendela dengan beberapa fungsi seperti berikut :

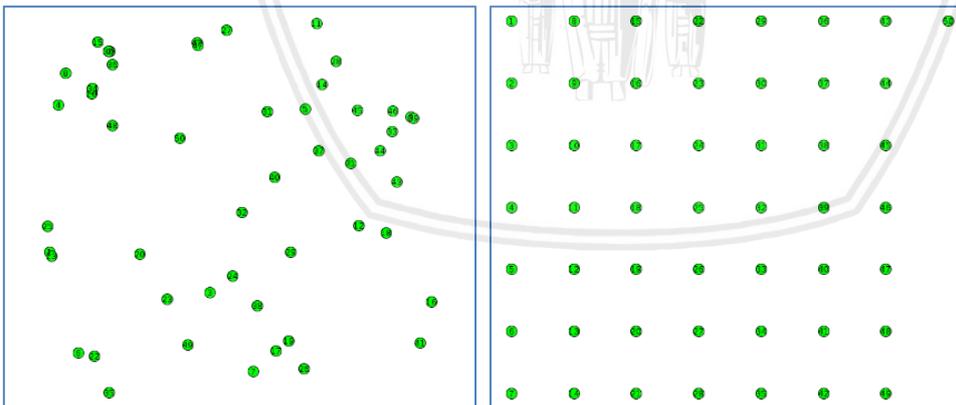
- *Network* – menampilkan lokasi setiap *node* pada jaringan, menampilkan status *mote* identitas, alamat, LED, dll.
- *Simulation Control* – Untuk menjalankan, memberhentikan, atau memuat ulang simulasi.

- *Notes* – Untuk menyimpan catatan pada simulasi.
- *Mote output* – Menampilkan keluaran antar-muka serial setiap *mote*.
- *Timeline* – Menampilkan garis waktu setiap kejadian pada jaringan.
- Antar muka lain seperti *Breakpoints*, *Radio Messages*, *Script Editor*, *Buffer View*, dan *Mote Duty-cycle*.

Mote yang disimulasikan pada *Cooja* merupakan hasil kompilasi dan eksekusi sistem Contiki yang sebelumnya didesain untuk *native-platform*. *Cooja* memuat sebagai sebuah *shared-library* menggunakan Java Native Interfaces. Beberapa jenis node dapat dijalankan dalam sebuah lingkungan simulasi yang sama, sehingga dapat mewakili sebuah jaringan yang heterogen. *Cooja* melakukan kontrol dan analisa terhadap sistem ContikiOS melalui beberapa fungsi. Misalnya, simulator menginformasikan sistem ContikiOS untuk menangani sebuah *event* atau menarik seluruh memori pada sistem ContikiOS untuk keperluan analisa.

2.9. Penempatan dan Topologi jaringan sensor

Terapat dua macam cara penempatan fisik WSN berdasarkan lokasi pemasangannya, dua cara seperti pada Gambar 2.11 tersebut adalah cara penempatan *random* and *lattice* (Yang et al., 2010). Pada penempatan *lattice*, sensor ditempatkan dengan jarak yang seragam untuk pengukuran ideal. Penempatan *node* secara *random* dibutuhkan untuk mendapat kondisi lingkungan yang lebih mirip dengan kondisi di lapangan.



Gambar 2.11 Penempatan *node* dengan metode *random* and *lattice*.

Topologi *wireless sensor network* pada penelitian ini adalah topologi *tree* karena ditentukan sebuah *node* yang berfungsi sebagai hilir komunikasi yang juga berfungsi sebagai *base-station*. Pada topologi *tree node-node* sensor diatur dalam susunan secara hierarki sehingga terdapat tiga macam *node*, yaitu *sensor-node*, *relay-node*, dan *base-station*. *Base-station* mengumpulkan informasi dari beberapa *sensor-node*, sedangkan *relay-node* selain berfungsi sebagai *sensor-node* juga berfungsi meneruskan informasi ke *base-station*.

2.10. Performa pada WSN

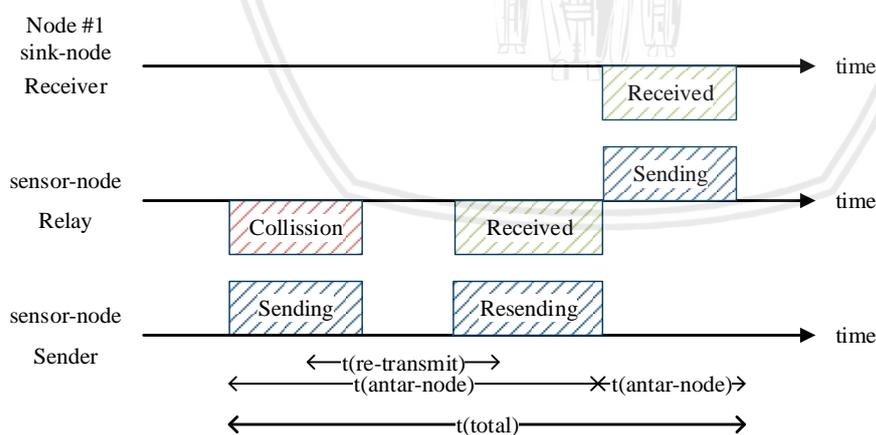
Cotrim (Cotrim & Kleinschmidt, 2017) menjelaskan beberapa parameter yang mempengaruhi kualitas pada jaringan sensor adalah efisiensi penggunaan energi, *latency*, dan keberhasilan pengiriman paket.

2.10.1. Konsumsi Energi

Pengukuran konsumsi energi pada WSN dapat dilakukan dengan menghitung waktu dan daya listrik yang dibutuhkan *node* pada mode *listen*, *transmit*, *receive*, *sleep*, dan ketika proses pengambilan data oleh sensor. Pengukuran diambil dalam satu siklus pengiriman. Lama siklus tersebut dapat tergantung pada persentase *duty-cycle* yang berbeda-beda pada setiap metode. Pada metode *on-demand* konsumsi energi dapat meningkatkan konsumsi energi karena ada tambahan daya yang dibutuhkan untuk modul tambahan dalam mode *sleep*, *transmit*, dan *receive*, untuk itu pemilihan modul tambahan yang hemat energi sangat menentukan.

2.10.2. Latency

Latency merupakan jarak waktu pengiriman paket data dari pengirim ke penerima. Semakin tinggi nilai *latency* tersebut maka semakin tinggi resiko kegagalan komunikasi. Faktor yang menyebabkan meningkatnya *latency* pada sebuah komunikasi data antara lain adalah jarak antara *sender* dan *receiver*, faktor alam yang terjadi pada media jaringan, *noise*, atau karena konfigurasi jaringan.



Gambar 2.12 Latency antar node

Latency pada komunikasi antar *node* (Gambar 2.12) dapat diidentifikasi dengan menghitung waktu kirim oleh *sender* pada *application-layer* (tsa) sampai dengan waktu terima oleh *reciever* pada *application-layer*, sehingga mencakup waktu kirim dan terima pada *radio-layer* dan waktu *delay* karena pengiriman ulang akibat *collision*.

2.10.3. Keberhasilan Pengiriman Paket

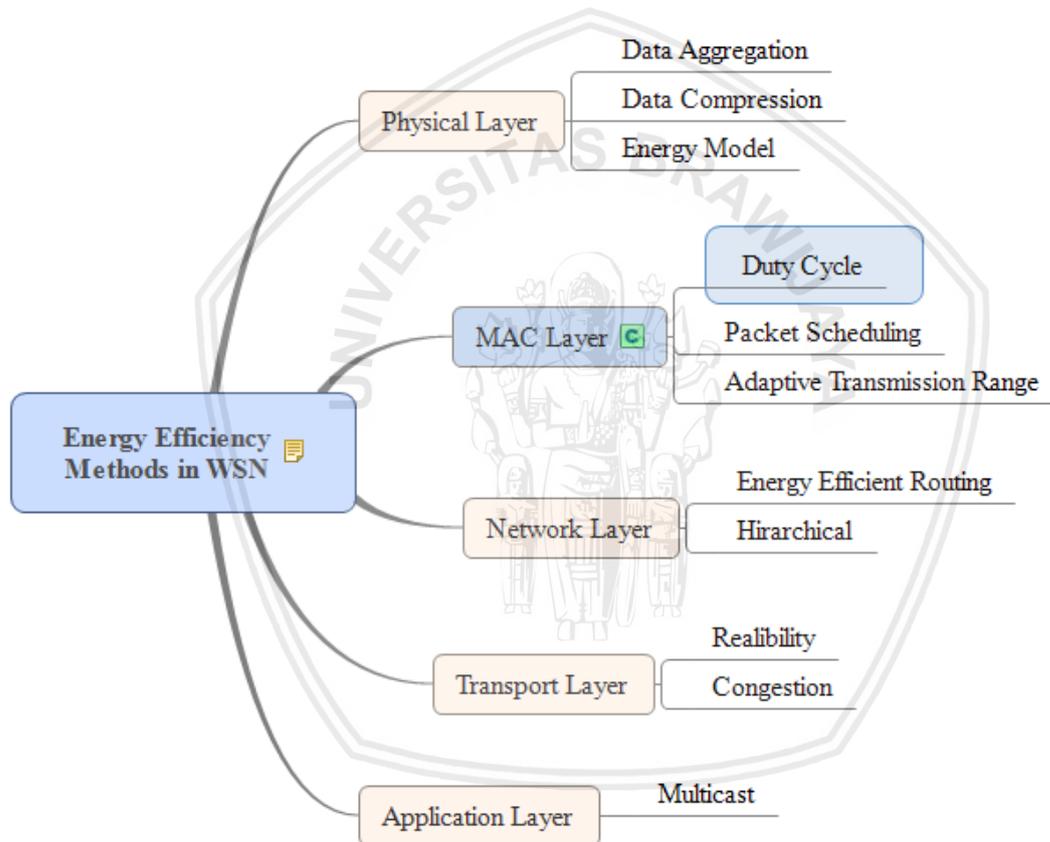
Keberhasilan pengiriman Paket/*Packet delivery ratio*(PDR) adalah perbandingan antara paket yang berhasil diterima dengan keseluruhan jumlah paket yang dikirimkan melalui jaringan. Jika pada jaringan terjadi kongesti dan tidak adanya mekanisme pengaturan yang baik, maka paket bisa mengantre di sisi pengirim, dan tidak akan pernah memasuki jaringan. Paket tersebut tidak mempengaruhi *throughput*, karena tidak pernah terkirim. Untuk itu pengujian PDR diperlukan pada WSN untuk mengetahui permasalahan yang ditimbulkan akibat *throughput* yang buruk.



BAB III

KERANGKA KONSEP PENELITIAN

Disiplin ilmu pada penelitian ini adalah efisiensi energi pada WSN, seperti pada Gambar 3.1 yang menunjukkan klasifikasi mekanisme efisiensi energi. Penghematan energi dapat dilakukan pada komponen-komponen perangkat keras pada *physical-layer*, atau dengan cara penjadwalan dan pengaturan jarak transmisi yang dinamis pada *media-access (MAC) layer*, atau metode metode lain pada *network-layer*, *transport-layer*, maupun *application layer* (Anastasi, Conti, Di Francesco, & Passarella, 2009).



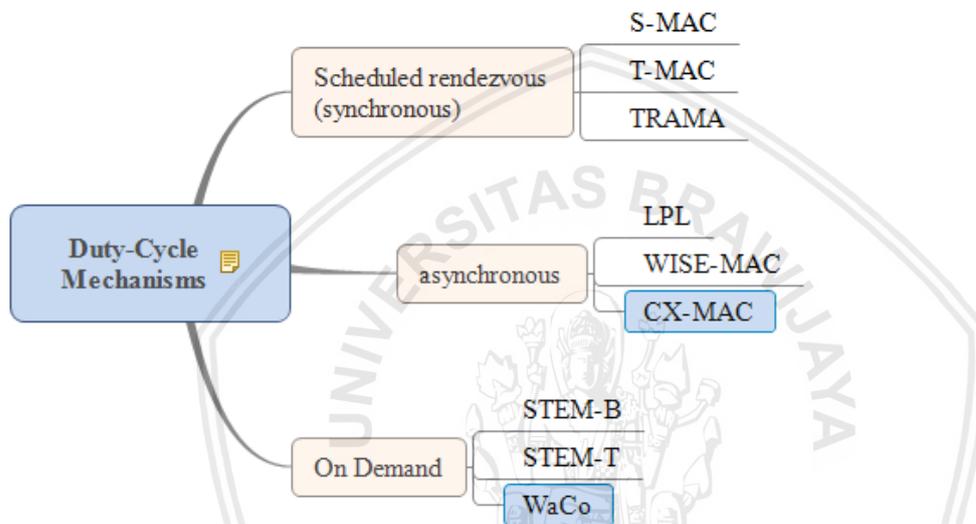
Gambar 3.1 Kerangka penelitian.

Metode *duty-cycle* pada lapisan *MAC* pada Gambar 3.1 (dibingkai dengan warna biru) merupakan metode penghematan konsumsi energi dengan menjadwalkan berapa lama sebuah *sensor-node* melakukan transmisi (t_x), dan berapa lama *sensor-node* tersebut harus mengaktifkan modul radio *receive* (t_r). Kemudian *sensor-node* tersebut juga dijadwalkan untuk bekerja dalam mode *sleep* (t_s) sehingga mengonsumsi energi yang rendah sampai *node* tersebut aktif karena *timer* atau mendapatkan sinyal *interrupt* dari sistem lain (t_{mcu}). Semakin



lama sensor bekerja dalam mode *sleep* (t_s), *sensor-node* akan dapat bekerja dalam waktu yang lebih lama, sampai sumber energi yang dimiliki habis.

Duty-cycle dipilih karena metode ini mempunyai dampak signifikan terhadap penghematan energi (Zhao & Wu, 2010)(Uwase et al., 2017). Berdasarkan mekanisme kerjanya, metode-metode tersebut dapat dikelompokkan menjadi tiga seperti pada Gambar 3.2. Penelitian mengenai mekanisme *Scheduled Rendezvous* atau *synchronous* antara lain adalah S-MAC (Ye et al., 2002) dengan penjadwalan tetap, T-MAC (Wang & Liu, 2007) dengan penjadwalan adaptif, dan TRAMA dengan pemilihan skema terdistribusi (Tay et al., 2004).



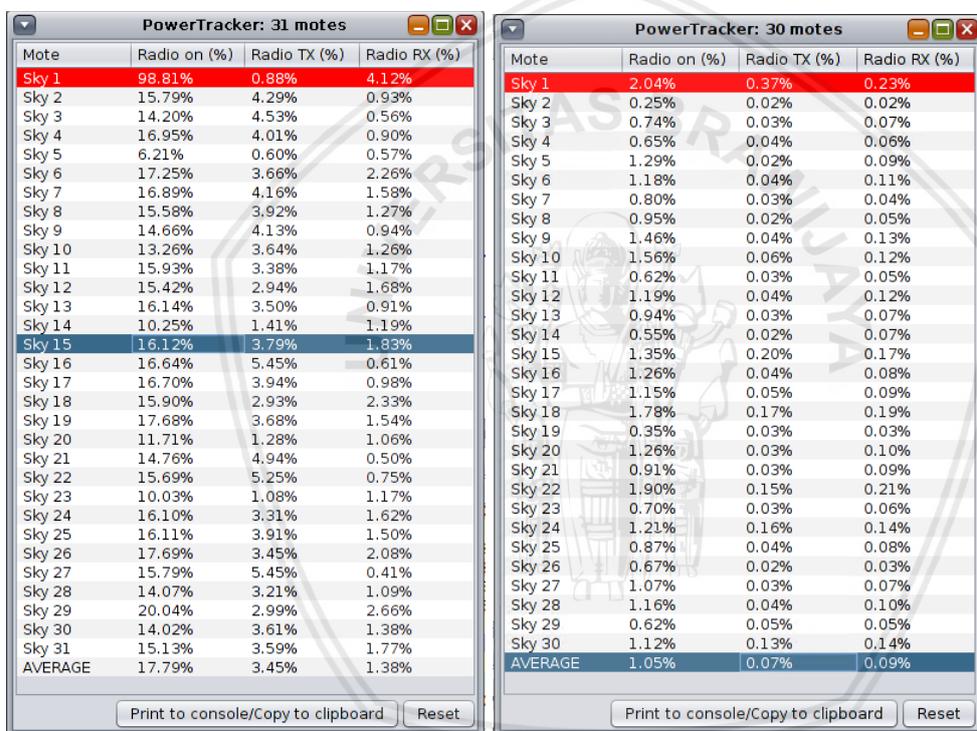
Gambar 3.2 Klasifikasi teknik *duty-cycle* dan contohnya

Mekanisme *asynchronous* yang paling sederhana adalah mekanisme LPL (El-Hoiydi, 2007) dengan menggunakan *preamble sample* tetap, WISE-MAC (Decotignie & Hernandez, 2005) dengan mengurangi *preamble sample*, CX-MAC (Buettner et al., 2006) dengan mengurangi *preamble sampling* dengan mekanisme pengiriman paket sama secara periodik dan menambahkan informasi alamat tujuan. Penelitian mengenai mekanisme *on-demand* antara lain adalah STEM-B (Schurgers, Tsiatsis, Ganeriwal, et al., 2002) yang menggunakan radio tambahan sebagai *beacon*, STEM-T (Schurgers, Tsiatsis, & Srivastava, 2002) yang memanfaatkannya sebagai *tone*, dan **waco** yang juga menggunakannya sebagai *tone* (Piyare, Istomin, et al., 2017). Beberapa metode mempunyai karakteristik masing masing. Berikut ini adalah perbandingan metode berdasarkan parameter penghematan *latency*, energi, dan kerumitannya.

Tabel 3.1 Perbandingan metode penjadwalan

No	Duty-cycle	Kategori	Cara Kerja	Latency	Energi	Mudah
1	SMAC (Ye, 2002)	Synchronous	Fixed duty-cycle, virtual cluster	-	-	++
2	TMAC (Wang, 2007)	Synchronous	Adaptive duty-cycle, over hearing	-	+	++
3	LPL (El-Hoiydi, 2007)	Asynchronous	Fixed preamble sampling	-	+	++
4	WiseMAC (Decotigni, 2005)	Asynchronous	Minimized preamble sampling	+	+	+
5	cxmac (Buettner, 2006)	Asynchronous	Sort preamble with target addrs	+	++	-
6	STEM (Schurgers, 2002)	On-demand	dual radio, broadcast a beacon	++	+	++
7	STEM-T (Schurgers, 2002)	On-demand	dual radio, tone by sender	++	+	++
8	waco (Piyare, 2017)	On-demand	dual radio, tone by receiver	++	++	-

Pada Tabel 3.1 parameter penghematan energi dan *latency* menjadi perhatian utama sesuai dengan survei yang dilakukan Khajuria (Khajuria & Gupta, 2015), sedangkan faktor kemudahan dapat menentukan biaya atau pemilihan jenis *hardware* pada saat implementasi.



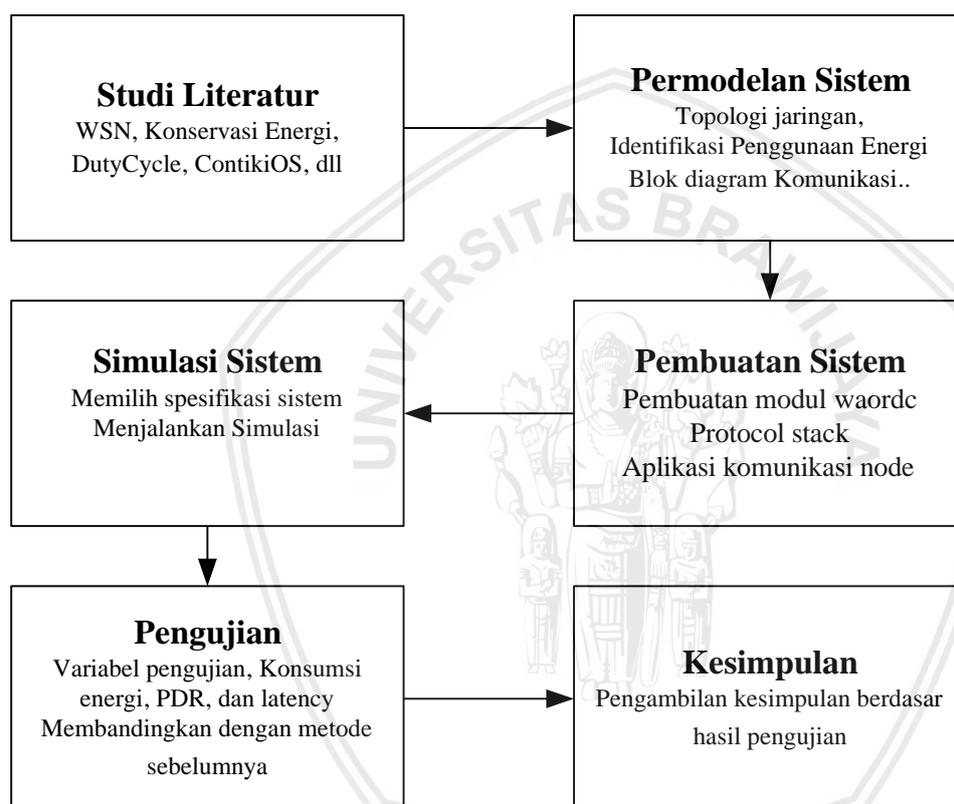
Gambar 3.3 Hasil pengukuran waktu *sleep* antara **cxmac** (kiri) dan **waco** (kanan).

Perbandingan kedua metode tersebut telah diuji untuk mengetahui dalam penghematan energi dan *throughput* (Alabi, 2017). Seperti yang terlihat pada Gambar 3.3 dapat dilihat bahwa **cxmac** yang menggunakan metode *asynchronous* mampu menghemat rata-rata *duty-cycle* dari 10% sampai 98%. Hasil ini memberikan pengaruh signifikan pada konsumsi energi pada *nodes*. Sedangkan **waco** yang menggunakan metode *on-demand* secara lebih efektif dibandingkan **cxmac** dengan melakukan *duty-cycle* sampai 1%. Meskipun demikian, pengukuran konsumsi energi pada radio tambahan (*Wake-up Radio/WuR*) perlu diperhatikan pada metode yang dilakukan **waco**.

BAB IV

METODOLOGI PENELITIAN

Untuk memastikan penelitian berjalan sesuai dengan kerangka konsep penelitian, maka dibuat alur langkah penelitian yang akan dilakukan. Dilakukan studi literatur mengenai efisiensi energi dan berbagai dasar teori yang menunjang. Kemudian dilakukan pemodelan, simulasi dan pembuatan sistem, dilanjutkan dengan pengujian dan pengambilan kesimpulan. Urutan metodologi yang dilakukan seperti pada Gambar 4.1.



Gambar 4.1 Alur langkah penelitian

4.1. Studi Literatur

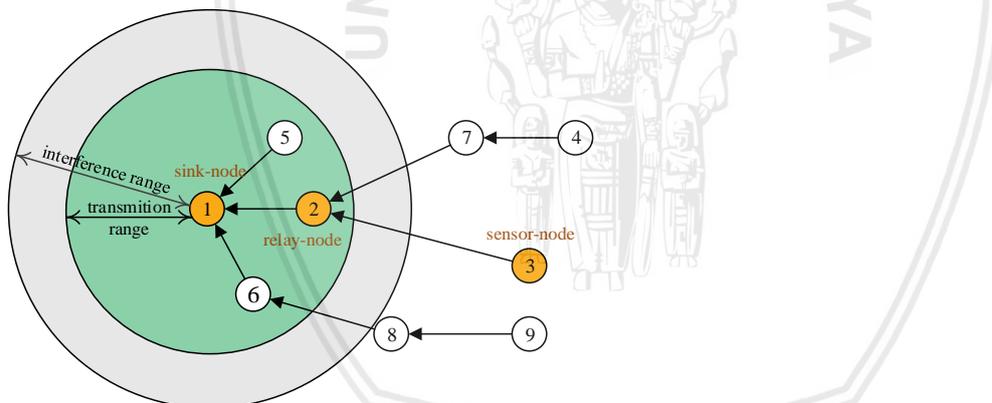
Tahap pertama adalah melakukan studi literatur dengan mengumpulkan dan mempelajari jurnal, buku, *datasheet*, atau publikasi lain yang diperoleh dari halaman web penyedia jurnal, perpustakaan, dan sumber lain mengenai WSN, efisiensi energi, *sleep scheduling*, dan lain-lain yang dibutuhkan dalam penyusunan laporan tesis. Studi literatur diperlukan untuk mendapatkan dasar teori dan merancang kerangka konsep penelitian yang berkaitan dengan penelitian yang dilakukan.

4.2. Pemodelan Sistem

Pada subbab ini dibahas bagaimana topologi yang digunakan, bagaimana peta konsumsi energi yang ada pada sebuah *sensor-node*, serta bagaimana komunikasi dalam jaringan WSN. Pemodelan sistem dilakukan untuk memperoleh gambaran sistem WSN dan rumusan konsumsi energi pada sebuah *node*. Dengan dilakukan pemodelan sistem didapatkan parameter-parameter yang diperlukan untuk keperluan penelitian dan analisa.

4.2.1. Topologi Jaringan

Topologi yang digunakan dalam penelitian ini mengacu pada penempatan sensor secara acak. Pada penempatan node, jarak antar node dibatasi berdasarkan daya pancar radio dan sensitivitas *receiver* yang digunakan, mengacu pada *datasheet* yang diberikan. Topologi yang digunakan seperti pada Gambar 4.2, terdapat minimal sebuah *node* yang berfungsi sebagai *relay-node* untuk menjembatani komunikasi *node* yang tidak dapat menjangkau *base-station* secara langsung. *Relay-node* pada penelitian ini akan berperan sebagai *node* yang akan diaktifkan oleh *sensor-node* kemudian mengaktifkan *node* lain setelahnya. Setelah radio utama aktif, *relay* akan meneruskan data ke-*node* berikutnya, begitu seterusnya.

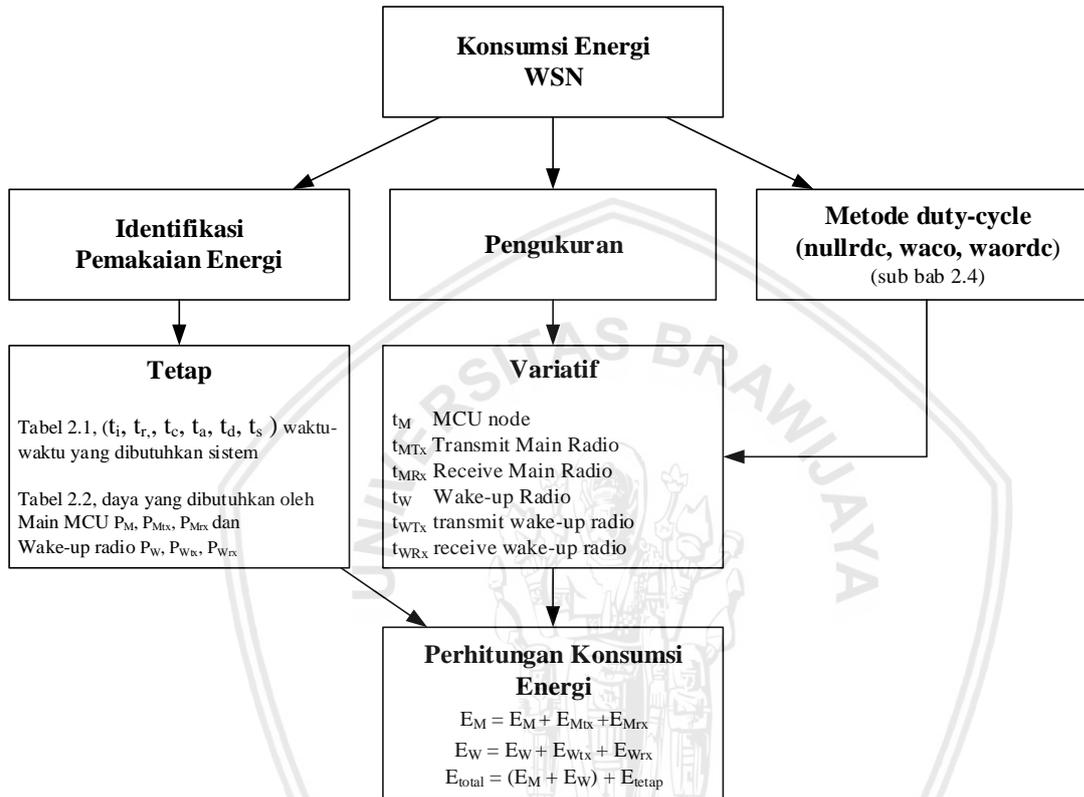


Gambar 4.2 Topologi tree dengan minimal sebuah relay-node.

Pada Gambar 4.2 digambarkan bahwa sebuah *node* memiliki dua lapisan jangkauan, *transmission-range* dan *interference-range*. *Transmission-range* merupakan jarak maksimal *node* dapat berhasil mengirimkan data, sedangkan *interference-range* merupakan area di mana *node* pengirim dapat mengganggu transmisi dari *node* pihak ke-3 yang melakukan pengiriman data (Chukwuka & Arshad, 2013). Supaya komunikasi dapat diterima dengan baik, peletakan *node* harus saling beririsan pada *transmission-range*.

4.2.2. Identifikasi Penggunaan Energi

Proses komunikasi radio (*transmit* dan *receive*) pada *node* menghabiskan energi paling banyak dibandingkan proses *data-sensing* and *processing*, meskipun dalam jumlah kecil proses *node* pada mode *sensing*, *sleeping*, dan *processing* juga menghabiskan energi. Masing-masing kondisi tersebut mempunyai profil penggunaan daya dan lama waktu yang berbeda-beda, seperti ditunjukkan pada Gambar 4.3 berikut ini :



Gambar 4.3 Identifikasi energi dalam sebuah siklus *duty-cycle* pada setiap *node*

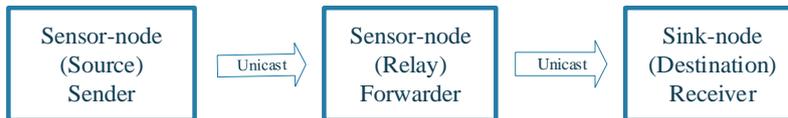
Konsumsi energi pada sebuah *node* dapat dijabarkan seperti pada Tabel 4.1. Komunikasi WSN merupakan jenis komunikasi *low power and lossy network*, sehingga sebuah *node* akan lebih sering dalam kondisi *idle* dibandingkan mengirimkan data. Penghematan energi dilakukan dengan mengurangi waktu aktif radio data dan MCU (t_M) dan waktu aktif radio *wake-up* (t_W) dengan mengorbankan keberhasilan dan kecepatan penyampaian paket.

Tabel 4.1 Penjabaran konsumsi energi pada WSN dengan mekanisme *duty-cycle*

No	State	Power(W)	Active period	E(Wh)
1	Idle	P_M	$t_M = (t_{cycle} - t_{sleep_m}) / t_{cycle} \%$	E_M
2	Transmit data	P_{MTx}	$t_{Mtx} = (t_{cycle} - t_{sleep_mtx}) / t_{cycle} \%$	E_{MTx}
3	Receive data	P_{MRx}	$t_{Mrx} = (t_{cycle} - t_{sleep_mrx}) / t_{cycle} \%$	E_{MRx}
4	Idle beacon	P_W	$t_W = (t_{cycle} - t_{sleep_w}) / t_{cycle} \%$	E_W
5	Sending beacon	P_{Wtx}	$t_{Wtx} = (t_{cycle} - t_{sleep_wtx}) / t_{cycle} \%$	E_{Wtx}
6	Waiting beacon	P_{Wrx}	$t_{Wrx} = (t_{cycle} - t_{sleep_wrx}) / t_{cycle} \%$	E_{Wrx}
Total				E_{total}

4.2.3. Blok Diagram Komunikasi pada WSN

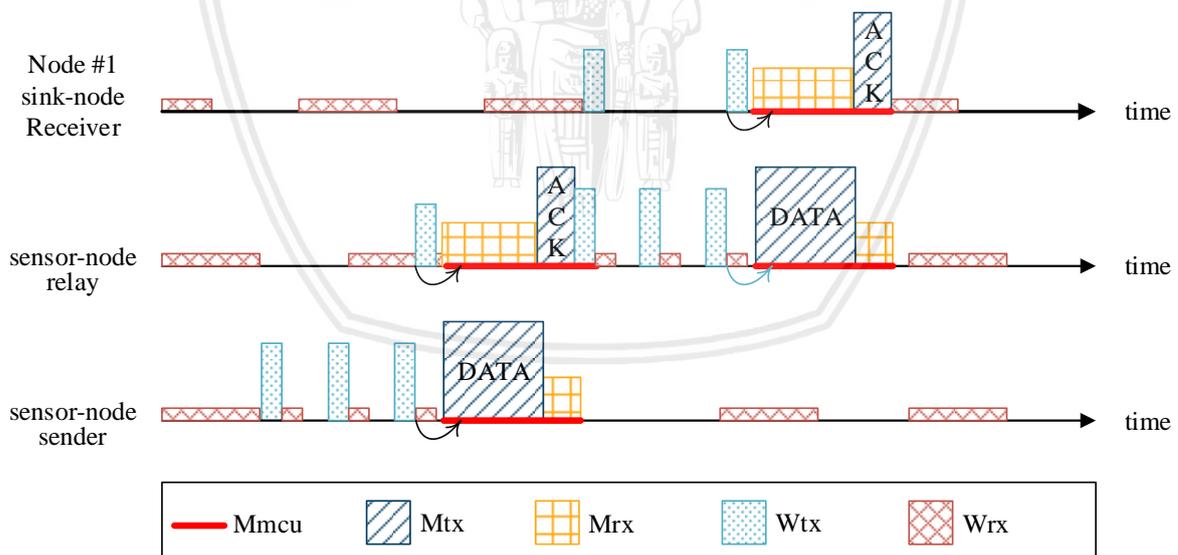
Dalam jaringan WSN untuk keperluan agrikultur, komunikasi dimulai oleh *sensor-node* yang telah mengumpulkan data dari sensor-sensor untuk kemudian diteruskan ke *sink-node* secara langsung atau melalui beberapa sensor. Pada Gambar 4.4 dapat dilihat proses pengiriman data dari *sensor-node* melalui sebuah *node*. *Node* perantara tersebut meneruskan paket ke *sink-node* untuk disimpan atau diteruskan ke pusat data.



Gambar 4.4 Blok diagram komunikasi antar *node* pada WSN

4.3. Pembuatan Sistem

Pada subbab ini dijelaskan cara kerja sistem dan pembuatan modul serta fungsinya ketika berfungsi sebagai *sensor-node*, *relay-node*, atau *base-station*. Modul yang diusulkan memperbaiki kekurangan yang ada pada metode *on-demand waco* (subbab 2.4.3 Gambar 2.8). dilakukan penjadwalan pada modul *Wrx* yang terus menerus aktif menggunakan teknik *asynchronous csmac* (subbab 2.4.2 Gambar 2.6). Sehingga *radio-duty-cycle* baru yang diusulkan dapat ditampilkan dalam diagram waktu seperti Gambar 4.5 berikut:



Gambar 4.5 Sistem *preamble-checking* pada *wake-up radio*.

Perubahan waktu *receive* pada radio tambahan diilustrasikan sebagai *Wrx*, jika sebelumnya pada metode *waco* modul tersebut selalu aktif, pada metode yang kami usulkan hanya aktif pada interval waktu tertentu dengan menggunakan mekanisme *asynchronous*. Selanjutnya metod yang diusulkan ini disebut *waordc* (*Wake-up Asynchronous on-demand radio duty-cycle*).

4.3.1. Protokol Stack

ContikiOS mengizinkan *network-layer* bekerja di atas beberapa macam protokol *mac layer* yang terbagi menjadi 3 *sub-layer*; MAC, RDC, dan Framer. Modul yang dibuat fokus pada RDC dengan nama **waordc**. Komunikasi antar node pada *network-layer* menggunakan *protocol-stack* Rime (Dunkels, 2007) dengan konfigurasi sebagai berikut:

- network layer : rime_driver
- framer : framer_802154
- rdc_driver : waordc_driver
- mac_driver : csma_driver
- radio_driver : cc2420_driver

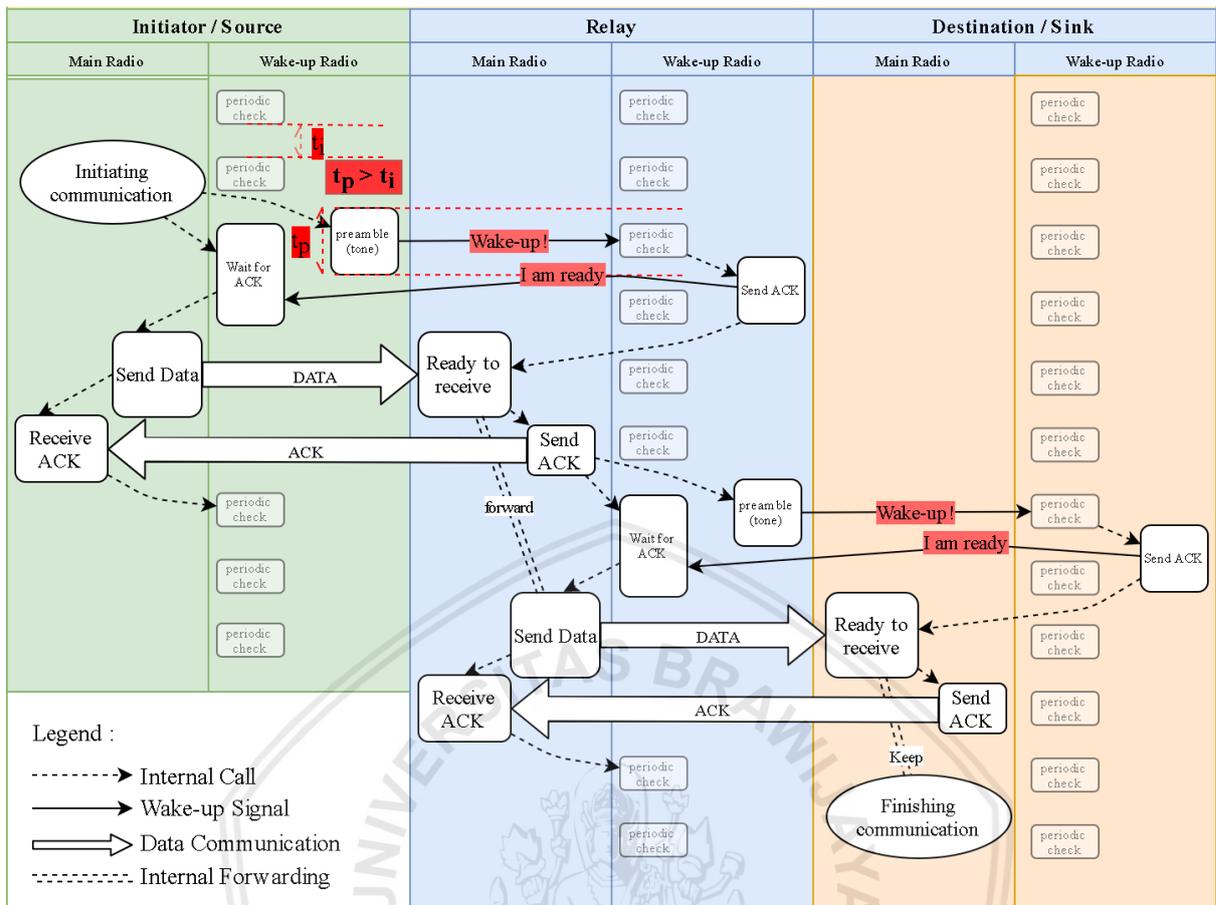
Untuk menghindari kehilangan paket pada saat pengiriman di waktu yang bersamaan, digunakan mekanisme CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) pada *mac-layer* yang mempunyai mekanisme sebagai berikut:

- Melakukan pengecekan medium sebelum mengirim.
- Membatalkan pengiriman apabila ada pengirim lain.
- Waktu pengiriman mengikuti penjadwalan oleh *rdc-layer*.
- Jumlah pengiriman-ulang mengikuti konfigurasi pada *Network-layer*.

Framer yang digunakan mengikuti standar 802.15.4 untuk kemudahan pembacaan *log* komunikasi dan menggunakan *driver* radio cc2420 dan modul *wake-up radio* msp430.

4.3.2. Komunikasi antar Node

Berdasarkan permodelan komunikasi tersebut, kemudian dibuat sebuah mekanisme pengujian pengiriman sebuah paket untuk mengukur lama waktu aktif radio ketika proses *idle*, mengirim *beacon*, proses transmisi dan proses menerima data. Pengujian dilakukan menggunakan pengiriman *unicast* dengan beberapa buah *node* sebagai *sender*, *relay*, dan *sink*. Pengujian menggunakan *protocol-stack* Rime yang mendukung mekanisme CSMA/CA untuk menjamin keberhasilan pengiriman paket dengan menghindari *collision*. Dengan pengujian ini, didapatkan penghematan energi pada masing-masing *node*.



Gambar 4.6 Activity-diagram modul asynchronous dan on-demand duty-cycle.

Seperti pada Gambar 4.6 komunikasi diawali oleh *sensor-node*. Hasil pengukuran parameter lingkungan menggunakan perangkat-perangkat sensor yang ada pada *sensor-node* kemudian dikirim dengan tujuan *sink-node* yang ada pada *base-station*. Dalam proses pengiriman data tersebut, *sensor-node* dapat langsung mengirim ke *sink-node* atau harus melalui *sensor-node* lain yang berfungsi sebagai *relay*. Proses yang terjadi pada jaringan mengikuti langkah-langkah sebagai berikut :

1. Proses awal

- a) Seluruh sensor bekerja sebagai *receiver* pada mode *stand-by*. Secara periodik, *node* mengaktifkan *receiver* pada *Wake-up Radio* untuk melakukan pengecekan adanya aktivitas di jaringan. Jarak antara pengecekan adalah t_i
- b) Apabila sebuah *node* akan mengirimkan sebuah paket, *sender* harus mengirimkan *preamble-tone*, sebuah sinyal sederhana yang berisi informasi tujuan menggunakan *transiver* yang ada pada *Wake-up Radio*. Lama waktu *preamble* sebesar t_p .
- c) Supaya *preamble-tone* dari *sender* tersebut dapat dibaca oleh *receiver*, kondisi $t_i < t_p$ harus terpenuhi.

2. Setelah mendapatkan data, *Sensor-node* mengawali proses komunikasi.

3. *Sensor-node* berlaku sebagai *sender* mengaktifkan *preamble-tone* dan memperpanjang waktu pengecekan periodik.
4. *Preamble-tone* kemudian mengaktifkan *node* yang berada di sebelahnya berdasarkan routing RPL.
5. *Node* yang berada di sebelahnya menerima *wake-up* kemudian melakukan pengecekan alamat.
6. *Node* tersebut kemudian mengaktifkan radio utama.
7. *Node* yang menerima sinyal *wake-up* tersebut kemudian membalasnya sebagai tanda bahwa *node* tersebut siap menerima data dari radio utama.
8. Setelah *sender* menerima konfirmasi, data sesungguhnya dikirim melalui radio utama
9. *Receiver* kemudian mengirimkan ACK melalui radio utama.
10. Setelah pengiriman data berhasil, *node* pengirim kembali pada mode *stand-by*.
11. *Receiver*, melakukan pengecekan alamat yang ada pada paket. Apabila paket tersebut sesuai, paket diteruskan seperti proses nomor 3 sampai 10.
12. Apabila alamat yang ada di dalam paket sesuai dengan alamat *receiver*, paket disimpan atau diteruskan ke pusat data untuk pemrosesan lebih lanjut.

Setelah permodelan komunikasi tersebut dibuat, kemudian dibuat sebuah modul *asynchronous duty-cycle* yang mengatur penjadwalan *Wake-up Radio*.

4.3.3. Pembuatan Modul waorc

Pada metode yang diusulkan, dilakukan perbaikan kekurangan yang ada pada metode *On-demand* menggunakan tambahan *Wake-up Radio* yang telah dilakukan Piyare (Piyare, Istomin, et al., 2017) dengan menambahkan sebuah modul yang menjadikan *Wake-up Radio* tersebut *sleep* secara periodik dengan sub-metode *asynchronous STEM-T* seperti yang pernah dilakukan Schurgers (Schurgers, Tsiatsis, & Srivastava, 2002).

Pembuatan modul baru tersebut dilakukan dengan melakukan modifikasi pada beberapa *source-code* ContikiOS. ContikiOS mengizinkan pengembang untuk melakukan penambahan modul baru untuk setiap lapisan *netstack*

4.3.3.1. PowerTracker untuk Radio Tambahan

Untuk memudahkan perhitungan persentase *duty-cycle* pada masing-masing *node*, dibutuhkan sebuah program *Powertracker*. Cooja hanya menyediakan *Powertracker* untuk pencatatan setiap *event* yang terjadi pada ContikiOS. *Event* bisa berupa berapa lama

waktu ketika sebuah sensor menyalakan modul radio, sensor, atau bekerja dalam *sleep mode*. Sedangkan untuk radio tambahan (WuR) perlu ditambahkan dengan melakukan pencatatan *event* pada modul **waordc**, kemudian ditampilkan pada sebuah jendela pengamatan *WurPowertracker* dengan menambahkan variabel radio tambahan pada berkas `tools/cooja/apps/wur_powertracker/java/WurPowerTracker.java`.

4.3.3.2. Mendaftarkan *protocol* Baru pada Netstack ContikiOS

Modul **waordc** yang dibuat perlu didaftarkan terlebih dahulu pada *protocol-stack* ContikiOS. **Waordc** bekerja pada *rdc layer*, yaitu berada di antara *physical layer* dan *media access layer*. *Rdc* harus dapat bekerja dengan protokol RADIO_cc2420 dan dapat melaksanakan perintah-perintah yang dikirim oleh protokol dari *media access layer* seperti CSMA. Konfigurasi *protocol* yang didaftarkan pada modul `netstack.h` sebagai berikut:

```
RADIO_cc2420.init();
RDC_waordc.init();
MAC_csma.init();
LLSEC_nullsec.init();
MAC_csma.init();
NETWORK_rime.init();
FRAMER_802154.init();
```

Konfigurasi tersebut kemudian ditambahkan pada berkas `core/net/netstack.h`. Fungsi `RDC_waordc.init()` akan memanggil header modul baru bernama `waordc.h`.

4.3.3.3. Menambahkan *Library* **waordc**

Modul **waordc** merupakan sebuah modul yang memungkinkan pengiriman paket harus diawali dengan mengaktifkan sinyal *preamble* dengan cara mengaktifkan W_{Tx} melalui fungsi `set_wuR_tx()` fungsi tersebut akan mengirimkan *preamble* berisi address tujuan sesuai instruksi yang perintahkan modul CSMA dari lapisan *media access layer*. Cara kerja modul ini sesuai dengan *pseudocode* berikut:

```
Init(Send_packet)
  Set rdc driver
input()
  if packetlen == acklen, ignore
  if not FRAMER.parse, error parsing
  if address not match, ignore
  detect duplicate
  read packet, forward MAC driver
  send ACK
Send_packet(), callback(input)
  copy link-address
  Set WuR Tx
  Clear WuR Tx
  Turn on radio
  Build message
  Transmit radio_tx
  if OK, then send packet
```

```

if Receiving send packet, pending
if Broadcast turn off, wait for rtimer
if Collision then turn radio off, wait for timer
if Noack then resend packet
default turn radio off

```

Modul ini mempunyai tiga fungsi utama, fungsi untuk memulai sistem operasi, fungsi *input* untuk membaca respons yang dikirimkan modul radio pada lapisan *radio/physical layer*, dan fungsi *send_packet()* yang dijalankan ketika *timer* mendapati waktu pengiriman data sensor atau ketika *node* mendapatkan paket yang harus diteruskan ke *sensor-node* berikutnya. Modul ini disimpan pada sebuah berkas *library waorc* di lokasi *core/net/mac/waorc.c*

4.3.3.4. Menambahkan *Driver waorc*

Supaya modul **waorc** dapat memberikan instruksi kepada radio tambahan yang dengan menggunakan *ultra low power mcu msp430*, dibuatkan sebuah *driver* baru dengan tambahan fungsi *sleep* selama sepertiga *cycle-time* dengan *pseudocode* sebagai berikut:

```

Set dc_timer cycle_time/3
Wur_init()
Set radio driver
Configure
  If sensor active
    Set interrupt pin Rx
    Select pin Rx
    Enable interrupt
  Select pin Rx
  Read pin Rx
  If address match, wait for td_timer
  Clear pin Rx
Wur_set_tx(), set Tx pin
Wur_clear_tx(), clear Tx pin
Wur_off(), Set sensor off, wait for ti_timer
Wur_on(), Set sensor on

```

Ketika W_{Rx} mendeteksi sebuah sinyal dengan alamat yang cocok dengan alamat *sensor-node*, modul *WuR* akan aktif lebih lama sampai waktu terpanjang pengiriman paket (t_d), namun jika selama waktu aktif tidak mendapatkan sinyal *wake-up*, modul *WuR* akan kembali *sleep* selama waktu interval (t_i).

4.3.3.5. Aplikasi *sensor_collect*.

Untuk mengumpulkan hasil *monitoring* dari tiap *sensor-node*, dibuat sebuah program *sensor_collect.c*. Program ini dijalankan pada *sensor-node* yang menjalankan sistem operasi *ContikiOS* dengan tambahan modul-modul sebelumnya. Fungsi-fungsi pada program ini didaftarkan sebagai sebuah *subprocess (thread)*. Fungsi *sensor_collect()* akan dijalankan terus-menerus selama *sensor-node* aktif, dan menjalankan fungsi *recv()* sesuai *pseudocode* berikut:

```

AUTOSTART_PROCESSES(sensor_collect);
recv(linkaddress, originator, seqno, hops)
    read packet message
    if not sink_node, run sensor_collect(message)
    print message
PROCESS_THREAD sensor_collect();
listen, callback(recv)
loop
    if interval_timer expired
        activate sensors
        build message
        deactivate sensors
        send data
    endif
endloop
ENDPROCESS_THREAD

```

Fungsi `recv()` dijalankan sebagai *callback*. Apabila *sensor-node* adalah *sink-node*, maka paket akan diterima kemudian menulis pada *log file*. Namun jika alamat tidak sesuai, paket akan diteruskan ke *sensor-node gateway* node tersebut.

4.4. Simulasi Sistem

Pada penelitian ini, akan dilakukan pengujian terhadap beberapa *sensor-node* yang tersebar secara acak. Jaringan sensor akan menemukan rute terbaiknya menggunakan protokol *routing* pada *protocol-stack* Rime (Cotrim & Kleinschmidt, 2017). Jumlah *node* dan posisi yang tersebut kemudian disimpan untuk kemudian digunakan sebagai *environment* pada pengujian berikutnya dengan kepadatan berbeda dan metode-metode sebelumnya. Dari hasil pengujian tersebut akan dilakukan perbandingan antara metode yang diusulkan dengan metode sebelumnya, serta bagaimana pengaruhnya pada kerapatan *node* yang berbeda.

4.4.1. Spesifikasi Sistem

Pengujian dijalankan menggunakan simulator Cooja yang berjalan pada sistem operasi linux dengan spesifikasi sebagai berikut :

- Virtualisasi = VMWare Workstation 14.00
- Sistem Operasi = Linux Ubuntu 14.04 LTS (instant-contiki 3.13.0-62-generic)
- CPU = Intel Core i7-7700 CPU @3.60GHz x8
- Memory = 8 GB

4.4.2. Menjalankan Simulasi

Simulasi dilakukan dengan *Cooja Simulator* menggunakan sistem operasi jaringan *ContikiOS* untuk *node* dengan.

Berikut adalah langkah untuk menjalankan simulasi sistem :

1. Menjalankan *Cooja* pada *virtual machine* Ubuntu. Membuat proyek baru, dengan jenis radio medium *Unit Disk Graph Medium (UDGM) : Distance Loss*.
2. Menambahkan *notes* dari *Sky notes CC2420* yang mendukung WuR. Program yang akan dimasukkan ke dalam *notes* ada pada file *waco/example-collect.c*
3. Buat file *waco/project.h*. Untuk dapat menjalankan konfigurasi yang berbeda pada setiap proyek, tambahkan baris berikut pada *Makefile* :

```
CONTIKI = /home/user/waco
CFLAGS += -DPROJECT_CONF_H=\"project-conf.h\"
include $(CONTIKI)/Makefile.include
```

kemudian buat file *waco/project-conf.h*.

4. Tambahkan baris berikut pada file *waco/project-conf.h* untuk masing-masing metode:
 - a. Untuk mengaktifkan metode **nullrdc**

```
#define NETSTACK_CONF_RDC nullrdc driver
```
 - b. Untuk mengaktifkan metode **cxmac**

```
#define NETSTACK_CONF_RDC cxmac driver
```
 - c. Untuk mengaktifkan metode **waco**

```
#define NETSTACK_CONF_RDC wurrdc driver
```
 - d. Untuk mengaktifkan metode **waordc** yang diusulkan (*Wake-up Asynchronous and On-Demand Duty-Cycle*)

```
#define NETSTACK_CONF_RDC waordc driver
```
5. *Compile contiki* dan muat ContikiOS ke dalam masing-masing *mote*.


```
$cd ../../examples/Waco/ && make clean && cd ../../tools/cooja
$ant run bigmem
```
6. Masukkan variabel dan konstanta yang digunakan dalam pengujian. (dijelaskan di bagian 4.5 Pengujian) .
7. Menjalankan simulasi *Cooja* dengan menekan tombol *start*
8. Mengamati dan merekam hasil pengujian.

4.5. Pengujian

Pengujian dilakukan dalam waktu operasi 21 menit simulasi untuk setiap metode. Menit pertama pada simulasi merupakan waktu *burn-in-time* supaya protokol *routing* dan MAC bekerja. Perubahan hasil pengujian dapat diamati pada 20 menit berikutnya dengan melakukan perekaman data untuk dilakukan analisa i) penggunaan energi, ii) keberhasilan pengiriman paket, iii) *latency*.

- Pengujian penggunaan energi. Untuk mengetahui tren penggunaan energi, dibutuhkan data hasil simulasi berupa rata-rata persentase waktu *duty-cycle* pada setiap *node*.

- Pengujian keberhasilan pengiriman paket. Pengujian ini dilakukan dengan membandingkan jumlah paket yang diterima sensor dengan keseluruhan paket yang dikirimkan.
- Pengujian *latency*. Pengujian ini diamati dengan menghitung rata-rata waktu yang dibutuhkan untuk mengirimkan paket dari sensor sampai diterima oleh *coordinator-node*.

Kedua langkah pertama tersebut dilakukan sebanyak empat kali pengujian dengan kerapatan *node* yang berbeda untuk mengetahui perbedaan performa dari masing-masing metode. Pada pengujian *latency* dilakukan penyesuaian waktu *wake-up* pada metode *cxmac* dan **waordc** sehingga diperoleh pengaruh *wake-up rate* terhadap performa jaringan seperti yang dilakukan penelitian sebelumnya (Uwase et al., 2017)

4.5.1. Variabel Global

Unit Disk Graph Radio Medium (UDGM) pada simulator Cooja mengabstrakkan transmisi radio sebagai sebuah lingkaran jangkauan transmisi. UDGM yang digunakan dalam simulasi adalah UDGM *Distance Loss*. Pada Tabel 4.2 ditampilkan beberapa variabel dan konstanta lain yang digunakan dalam penelitian ini mengacu pada simulasi yang dilakukan pada (Sampayo, Montavont, Prégaldiny, & Noel, 2018) dengan beberapa penyesuaian.

Tabel 4.2 Parameter-parameter pengujian

Parameter	Nilai
Durasi Simulasi	20 menit
Jumlah Node	3, 10, 25, 50
<i>Transmission range</i>	50 m
<i>Interference range</i>	100 m
Luas Area	100x100 m ²
<i>Packet rate</i>	2 paket / menit
<i>CXMAC channel check period</i>	125 ms
<i>MAC layer</i>	CSMA (Contiki version)
<i>Max-Retransmit CSMA</i>	3
<i>Network layer</i>	RIME
Mote	Sky mote
<i>Main Radio</i>	CC2420
<i>Desain Hardware Wake-up Radio</i>	(Piyare, Murphy, et al., 2017)
<i>Wake-up Radio MCU</i>	MSP430™ ULP MCU
<i>Radio medium model</i>	UDGM Distance Loss

Penyesuaian parameter simulasi adalah jumlah *node* sebanyak 3 dan 10 untuk merepresentasikan jaringan sensor yang renggang, sementara 25 dan 50 dapat mewakili kondisi jaringan *mesh* dengan kerapatan tinggi. Pembatasan 50 buah karena keterbatasan kemampuan

prosesor menjalankan simulasi. Nilai *paket-rate* pada Tabel 4.2 disesuaikan dengan pengujian PAN yang telah dilakukan sebelumnya (Muhammad, 2018).

4.5.2. Dataset Pengujian PAN

Pengujian dijalankan dalam sebuah simulasi pengumpulan data dari sensor *node* ke *sink-node* (*node* dengan ID 1) setiap 30 detik. Penggunaan energi akan dihitung berdasarkan hasil pengukuran yang pernah diimplementasikan pada *hardware* CC2420 menggunakan *wake-up* radio pada frekuensi sub-1GHz (Piyare, Istomin, et al., 2017). Hasil pengukuran piyare dkk. akan dijadikan acuan perhitungan besar energi yang dibutuhkan dalam sebuah *network*.

Untuk mengetahui mekanisme pengujian berupa pengiriman rentetan data dari sebuah *dataset* hasil pengamatan *environment* tanaman porang yang telah diambil sebelumnya. *Dataset* yang dikirimkan berupa hasil *monitoring* suhu, temperatur dan kelembapan tanah yang diambil dari lokasi penanaman Porang (Muhammad, 2018). Disiapkan 65617 data dengan format seperti pada Tabel 4.3.

Tabel 4.3 Hasil monitoring lingkungan untuk penanaman porang

id	date	ammo	temp	hum
MC-01	8:37:15 AM	82.74	28.16	71.95
MC-01	8:36:15 AM	82.56	28.09	72.44
MC-01	8:35:15 AM	82.36	27.94	74.02

Sumber: (Muhammad, 2018)

Data yang kami kumpulkan sebelumnya merekam data-data tersebut selama tiga bulan masa akhir penanaman. Dengan data seperti pada Tabel 4.3 kami akan menjalankan simulasi pengiriman paket dengan metode pengumpulan, masing-masing *node* mengirimkan data menuju sebuah *node* induk (*sink-node*) (Muhammad, 2018).

Dalam pengujian metode *On-demand and Asynchronous duty-cycle* yang kami usulkan ini, pengujian *latency* merupakan jeda waktu yang terjadi pada lingkup PAN. Keberhasilan pengiriman paket juga diamati dari berapa paket yang sampai pada *node* induk (*sink-node*) dalam satu area PAN yang sama. Pengujian dilakukan dengan menjalankan program pengumpulan hasil pengukuran sensor yaitu *contiki-collect*, aplikasi yang sama dengan yang digunakan oleh piyare (Piyare, Murphy, et al., 2017).

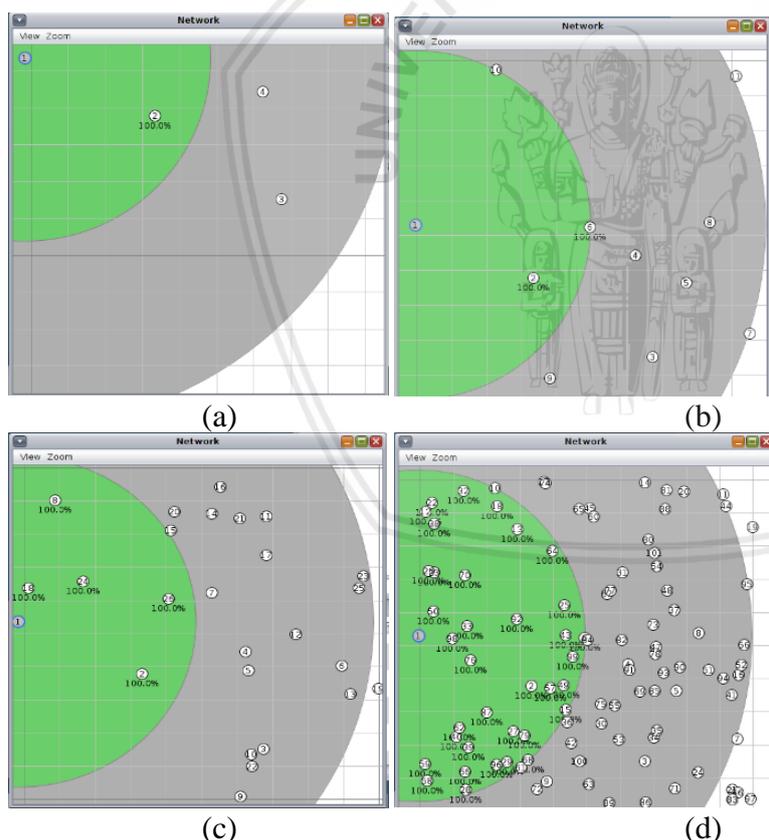
BAB IV

HASIL DAN PEMBAHASAN

Pada bab 5 ini ditampilkan hasil pengujian-pengujian yang telah disimulasikan sebelumnya. Pada bab ini akan ditampilkan penempatan *node* yang membentuk topologi jaringan *tree* yang akan disimulasikan. Kemudian ditampilkan pula hasil pengujian konsumsi energi, keberhasilan pengiriman paket, serta *latency* rata-rata yang terjadi pada pengiriman paket dari masing-masing *node* ke *sink-node* (*sink* dengan ID 1).

5.1. Hasil Pengujian

Penentuan lokasi *node* yang dilakukan secara acak dengan 4 kepadatan yang berbeda sesuai penjabaran simulasi sistem pada subbab 4.4. Pada Gambar 5.1 tampak tampilan jendela simulasi Cooja Network yang menampilkan peletakan *node-sensor*:



Gambar 5.1 Penempatan 3, 10, 25, dan 50 *node*

Peletakan *sensor-node* pada Gambar 5.1 dilakukan secara acak dengan kerapatan 3 *sensor-node* (a), 10 *sensor-node* (b), 25 *sensor-node* (c), dan 50 *sensor-node* (d) pada luas area $100 \times 100 \text{ m}^2$. Semakin renggang kerapatan jaringan WSN, maka semakin sedikit interferensi pada kanal jaringan. Interferensi dari *sensor-node* lain dapat menyebabkan pengiriman paket

repository.ub.ac.id

tertunda atau bahkan gagal. Di sisi lain, semakin padat kerapatan jaringan WSN, *sensor-node* akan memiliki opsi lain dalam penentuan rute. Pengujian menggunakan metode **nullrdc**, **cxmac**, **waco**, dan **waordc** dilakukan pada 4 lokasi penempatan tersebut.

5.2. Pengujian Penggunaan Energi

Untuk mengetahui konsumsi energi dalam masing-masing *node*, dilakukan pengujian selama 21 menit di mana menit pertama tidak dilakukan pengiriman data sampai *node* mengetahui mana rute yang tepat untuk mengirimkan data sampai kepada *sink-node*. Pengujian dihentikan setelah 20 menit pengambilan data karena keterbatasan pengalamatan *memory* pada sistem operasi 32 *bit* dalam menjalankan simulasi dengan jumlah *node* sebanyak 50.

Untuk mengetahui persentase penggunaan energi masing-masing *node* dalam sebuah jaringan WSN, dilakukan perincian waktu aktif masing-masing modul yang ada dalam sebuah *node*. Cooja dapat membaca *event* yang terjadi dalam ContikiOS, sehingga dapat ditampilkan waktu saat sebuah modul aktif, dan waktu saat modul tersebut mati atau waktu saat *sensor-node* dalam keadaan *sleep* atau *idle*.

5.2.1. Detail Waktu Aktif *Node* dan Modul Radio

Tabel 5.1 menunjukkan hasil rata-rata pengukuran waktu aktif masing-masing modul per-*node*. Lama waktu aktif merupakan persentase waktu aktif sebuah modul selama rentang waktu pengujian (20 menit). Pada metode **nullrdc** dan **cxmac**, W_uR , W_{tx} , dan W_{rx} ditampilkan selalu bernilai 0 karena kedua metode tersebut memang tidak menggunakan radio tambahan dalam pengiriman data. Sedangkan pada metode **waco**, W_uR bernilai mendekati 100% karena tidak menggunakan penjadwalan sama sekali pada radio tambahannya.

Metode yang kami usulkan mempunyai waktu aktif sebesar 67% dikarenakan dalam satu siklus radio tersebut dijadwalkan mati selama 1/3 waktu dan kembali aktif selama 2/3 waktu. Pada Gambar 5.1 juga menunjukkan data pengujian penggunaan energi dengan kerapatan *node* yang berbeda beda pada suatu luas area yang sama ($100 \times 100 \text{ m}^2$). Pada jaringan dengan kerapatan *node* yang sangat padat dapat menyebabkan adanya *collision* pada lalu-lintas data. *Collision* tersebut dapat menyebabkan pengiriman ulang paket sehingga menambah waktu aktif radio utama maupun radio tambahan. *Collision* juga dapat menyebabkan kegagalan pengiriman paket ketika pengiriman ulang paket mencapai batas *max-retransmit* sesuai konfigurasi pada subbab 4.5.1 Tabel 4.2.

Tabel 5.1 Hasil pengukuran waktu aktif selama pengujian 20 menit.

Total of nodes	Module	Duty-cycle %			
		nullrdc	cxmac	waco	waordc
3 nodes	MR	99.94	6.79	0.17	0.17
3 nodes	M _{tx}	0.02	0.11	0.03	0.03
3 nodes	M _{rx}	0.04	0.09	0.03	0.03
3 nodes	WuR	-	-	99.95	74.8
3 nodes	W _{tx}	-	-	0.01	0.01
3 nodes	W _{rx}	-	-	0.01	0.01
10 nodes	MR	99.94	7.23	0.41	0.41
10 nodes	M _{tx}	0.02	0.2	0.06	0.06
10 nodes	M _{rx}	0.12	0.12	0.05	0.05
10 nodes	WuR	-	-	99.96	76.24
10 nodes	W _{tx}	-	-	0.01	0.01
10 nodes	W _{rx}	-	-	0.02	0.02
25 nodes	MR	99.94	9.5	0.85	0.85
25 nodes	M _{tx}	0.03	0.53	0.06	0.06
25 nodes	M _{rx}	0.3	0.46	0.08	0.08
25 nodes	WuR	-	-	99.95	70.3
25 nodes	W _{tx}	-	-	0.01	0.01
25 nodes	W _{rx}	-	-	0.07	0.02
50 nodes	MR	99.94	7.07	2.65	2.65
50 nodes	M _{tx}	0.03	0.8	0.08	0.08
50 nodes	M _{rx}	0.68	0.5	0.23	0.23
50 nodes	WuR	-	-	99.96	73.5.
50 nodes	W _{tx}	-	-	0.01	0.01
50 nodes	W _{rx}	-	-	0.2	0.02

5.2.2. Konsumsi Energi Keseluruhan Node

Dari pengukuran persentase waktu aktif tersebut dapat diketahui rata-rata energi yang dibutuhkan oleh masing-masing *node* berdasarkan pengukuran daya untuk masing-masing modul yang telah dilakukan piyare sebelumnya, diketahui kebutuhan energi keseluruhan dengan merujuk daya yang dibutuhkan masing-masing modul pada subbab 2.5 Tabel 2.2 sehingga dapat diketahui kebutuhan energi yang dibutuhkan dalam sebuah 20 menit pengiriman data. Energi ditampilkan pada Tabel 5.2 melalui persamaan sebagai berikut:

$$E = \sum_{i=1}^n (\text{Persentase aktif}) * 20/60 * (\text{wattage_modul}_i)$$

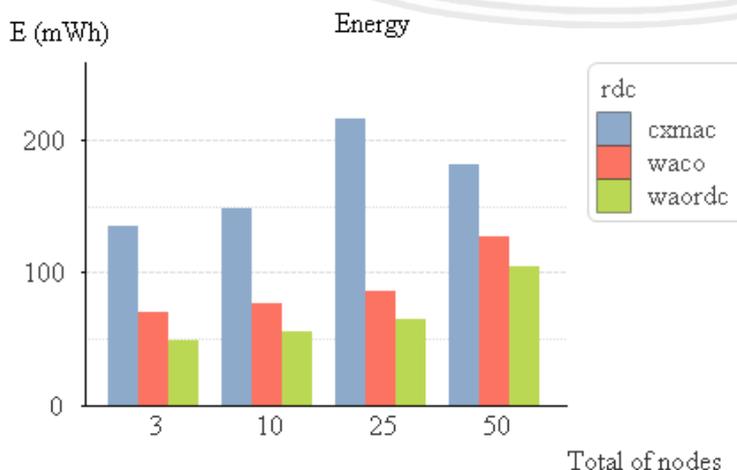
Tabel 5.2 *Detail kebutuhan energi per-node dalam 20 menit untuk 3 node*

Module	Power per-module (watt)	Duty-cycle %				Energy (watt hour)			
		method				method			
		nullrdc	cxmac	waco	waordc	nullrdc	cxmac	waco	waordc
MR	56.400	99.94	6.79	0.17	0.17	1878.872	127.652	3.196	3.196
M _{tx}	52.200	0.02	0.11	0.03	0.03	0.724	3.982	1.086	1.086
M _{rx}	56.400	0.04	0.09	0.03	0.03	1.504	3.384	1.128	1.128
WuR	1.944	-	-	99.95	67	-	-	64.768	43.416
W _{tx}	8.380	-	-	0.01	0.01	-	-	0.216	0.034
W _{rx}	1.440	-	-	0.01	0.01	-	-	0.033	0.011

Tabel 5.2 menunjukkan perhitungan energi per-*node* selama 20 menit. Konsumsi energi pada modul M_{tx} dan M_{rx} dihitung dengan menjumlahkan daya MR ketika *Idle* dengan daya yang digunakan ketika melakukan *transmit* data atau *receive* data. Sedangkan konsumsi energi pada modul W_{tx} dan W_{rx} dihitung dengan menjumlahkan daya WuR ketika *Idle* dengan daya yang digunakan ketika mengirimkan preamble paket atau ketika mengaktifkan *probe*. Pada metode **nullrdc** dan **cxmac** tidak dilakukan perhitungan konsumsi energi pada modul WuR, W_{tx} dan W_{rx} karena hanya menggunakan satu radio (MR) untuk melakukan komunikasi. Detail perhitungan energi untuk jumlah sensor 10, 25, dan 50 dilampirkan pada Lampiran 1.

5.2.3. Perbandingan Konsumsi Energi

Berdasarkan perhitungan penggunaan energi selama 20 menit, dapat dibandingkan kebutuhan energi antar masing-masing *node* dengan parameter uji yang sama, sehingga dapat diketahui bahwa metode **waordc** yang diusulkan membutuhkan energi lebih sedikit dibandingkan dengan 3 metode yang lain.



Gambar 5.2 Kebutuhan energi rata-rata sebuah sensor *node* selama 20 menit pengiriman

Pada Gambar 5.2 tampak penerapan metode *duty-cycle* dapat menghemat energi dalam sebuah *node*, dengan menerapkan metode *on-demand* dengan penambahan modul *wake-up* radio sebagai *trigger* untuk mengaktifkan *node* tujuan dapat mengurangi penggunaan energi menjadi 68,39 mWh, lebih baik dari metode *waco* yang sebesar 90,16 mWh dari rata-rata penggunaan energi keseluruhan 1890,49 mWh selama 20 menit jika tanpa dilakukan penjadwalan.

Tabel 5.3 Perbandingan konsumsi energi berdasarkan kepadatan node di area 100x100m²

Total of nodes in 100x100m ²	Energy (mWh)			
	nullrdc	cxmac (asynchronous)	waco (on-demand)	waordc (hybrid)
3 nodes	1881.10	135.02	70.43	48.87
10 nodes	1884.11	147.68	76.82	55.23
25 nodes	1891.24	215.08	86.37	64.63
50 nodes	1905.53	180.68	127.01	104.84
Average	1890.49	169.61	90.16	68.39

Tabel 5.3 menampilkan detail kebutuhan energi sebuah *node* dengan kepadatan berbeda dalam rentang waktu pengiriman data 20 menit dengan *traffic* pengiriman sebanyak 2 paket per menit dalam satuan *mili watt hour*(mWh). Kebutuhan energi tersebut adalah untuk mengaktifkan modul perangkat radio utama dan radio *wake-up*.

5.3. Pengujian Keberhasilan Pengiriman Paket (PDR)

Pada pengujian keberhasilan pengiriman paket, dilakukan perbandingan paket yang telah dikirimkan oleh *node* sensor dengan jumlah paket yang sampai pada *sink-node* yang dikirim secara acak dalam rentang waktu 30 detik. Faktor-faktor yang dapat mengakibatkan paket hilang antara lain adalah terjadinya *collision*, semakin rapat jaringan sensor, maka semakin sering terjadi *collision* karena banyak *node* lain di dalam *interference-range*.

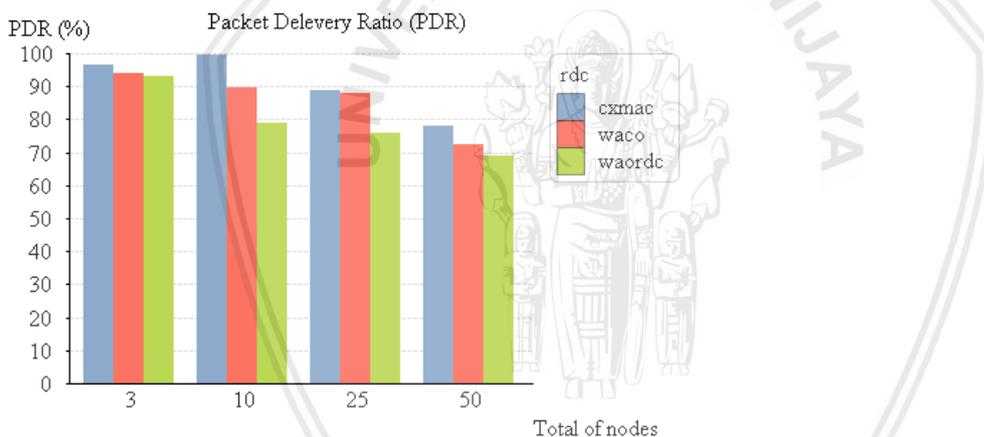
Tabel 5.4 Perbandingan keberhasilan pengiriman paket masing-masing metode

Total of nodes in 100x100m ²	PDR (%)			
	nullrdc	cxmac (asynchronous)	waco (on-demand)	waordc (hybrid)
3	100.0	96.7	94.2	93.1
10	100.0	100.0	89.8	79
25	100.0	88.8	87.8	75.7
50	100.0	78.0	72.6	69.2

Pada pengujian keberhasilan pengiriman paket, dilakukan perbandingan paket yang telah dikirimkan oleh *node* sensor dengan jumlah paket yang sampai pada *sink-node* yang

dikirim secara acak dalam rentang waktu 30 detik. Faktor-faktor yang dapat mengakibatkan paket hilang antara lain adalah terjadinya collision. Semakin rapat jaringan sensor, maka semakin sering terjadi collision karena banyak node lain di dalam interference-range. Pada pengujian keberhasilan pengiriman paket, dilakukan perbandingan paket yang telah dikirimkan oleh *node* sensor dengan jumlah paket yang sampai pada *sink-node* yang dikirim secara acak dalam rentang waktu 30 detik. Faktor-faktor yang dapat mengakibatkan paket hilang antara lain adalah terjadinya *collision*, semakin rapat jaringan sensor, maka semakin sering terjadi *collision* karena banyak *node* lain di dalam *interference-range*.

Tabel 5.4 menampilkan bahwa metode yang diusulkan memiliki tingkat keberhasilan pengiriman paket tidak sebaik metode yang lain, terutama ketika kepadatan *node* semakin tinggi. Akan tetapi, jika kepadatan *node* rendah sampai sedang, keberhasilan pengiriman paket tidak mengalami penurunan besar dibandingkan dengan metode *duty-cycle* lain yaitu **waco**, lihat Gambar 5.3.



Gambar 5.3 Grafik perbandingan keberhasilan pengiriman paket (PDR)

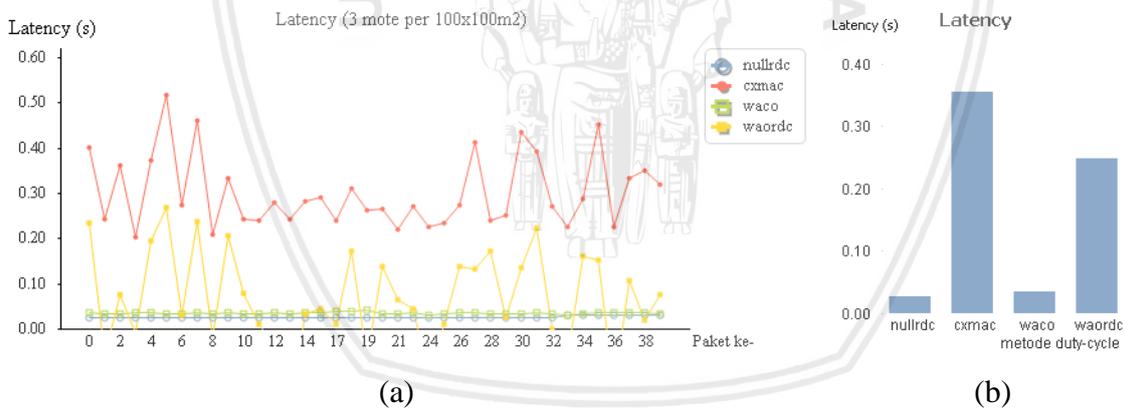
5.4. Pengujian Latency

Pada pengujian latency, selisih waktu pengiriman didapatkan dengan melakukan penulisan *log* dan waktu pada simulator pada simulator Cooja ketika sebuah node memulai proses pengiriman, kemudian dilakukan pencatatan *log* dan waktu pada simulator kembali ketika paket terbaca oleh *node* tujuan. Pada pengujian pertama, dilakukan pada jaringan dengan kepadatan rendah dengan merekam *latency* pada setiap paket yang dikirim oleh masing-masing *node*, kemudian dilakukan rata-rata nilai *latency* pengiriman paket dengan urutan paket sama, 10 paket pertama ditampilkan pada Tabel 5.5.

Tabel 5.5 Rata-rata *latency*

Sequence number	nullrdc	cxmac	waco	waordc
0	0.026	0.401	0.036	0.286
1	0.026	0.243	0.033	0.175
2	0.026	0.362	0.035	0.291
3	0.026	0.205	0.036	0.053
4	0.026	0.373	0.036	0.255
5	0.026	0.517	0.033	0.368
6	0.026	0.274	0.033	0.179
7	0.026	0.460	0.036	0.329
8	0.026	0.210	0.033	0.115
9	0.026	0.333	0.036	0.216

Penggunaan metode **cxmac** (*asynchronous*) memanfaatkan *preamble* yang mempunyai jeda waktu panjang, ketika sebuah node memulai untuk mengirimkan paket, sedangkan tidak ada *node* terdekat yang aktif, *node* pengirim akan menunggu cukup lama, sehingga berdampak pada *latency* seperti pada Gambar 5.4 (a) sedangkan metode **waordc** (on-demand) yang menggunakan mekanisme serupa mengatasinya dengan menurunkan *preamble-time* dengan jeda yang singkat. Pada grafik tersebut, metode **nullrdc** dijadikan sebagai acuan karena metode **nullrdc** pada kepadatan rendah memiliki *latency* paling kecil karena tidak terjadi



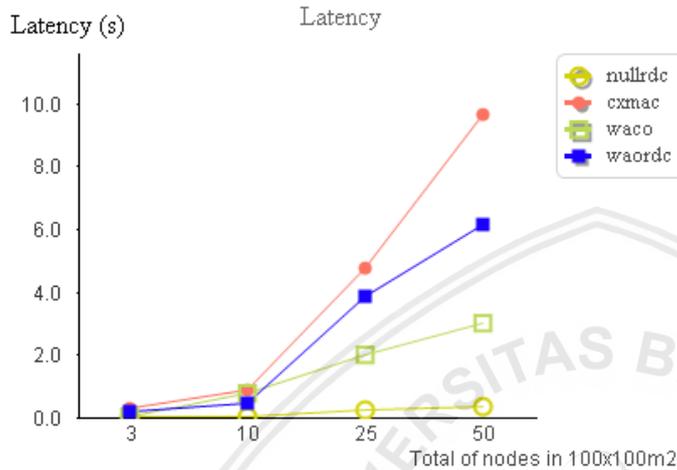
Gambar 5.4 *Latency* pengiriman paket dengan kepadatan node rendah

Gambar 5.5 (b) menunjukkan rata-rata seluruh *latency* pada semua urutan paket dengan kepadatan 3 *node* per 100x100 m². **Waordc** masih menampilkan performa yang lebih baik dibandingkan penjadwalan menggunakan metode *asynchronous* seperti **cxmac**.

Pada pengujian dengan kepadatan jaringan yang berbeda ditampilkan pada Tabel 5.6 menunjukkan bahwa metode **waordc** mempunyai kualitas lebih baik dibandingkan dengan mekanisme *asynchronous* **cxmac**:

Tabel 5.6 Perbandingan latency berdasarkan kerapatan node

method	Latency (s)		
	3 node	10 node	25 node
nullrdc	0.027	0.027	0.289
cxmac (asynchronous)	0.355	0.896	4.639
waco (on-demand)	0.035	0.806	2.031
waordc (hybrid)	0.120	0.447	3.782



Gambar 5.5 Perbandingan latency berdasarkan kerapatan jaringan

Gambar 5.5 menunjukkan perbandingan rata-rata lama waktu sebuah paket dikirim oleh sebuah sensor *node* sampai diterima oleh *node* tujuan (*sink-node*). Sehingga dapat diketahui bahwa terjadi peningkatan *latency* pada metode **waordc** lebih besar dibandingkan dengan metode **waco** dan lebih cepat dibandingkan dengan metode **cxmac**.

5.5. Pengaruh Mekanisme Penjadwalan Terhadap Kualitas Jaringan

Dengan membandingkan **nullrdc** sebagai acuan dasar terhadap tiga metode yang lain, maka dapat diketahui efisiensi energi yang dibutuhkan ketika menggunakan metode penjadwalan **cxmac** adalah sebesar 8,97% (169,61 mWh), metode **waco** sebesar 4,76% (90,16 mWh), dan **waordc** sebesar 3,61% (68,39 mWh) dari total energi yang dibutuhkan 1.890,49 mWh untuk mengaktifkan *sensor-node* dalam waktu 20 menit. Di sisi lain terjadi penurunan keberhasilan pengiriman data pada metode **cxmac** menjadi 90,88% dengan *delay* pengiriman 4,685 detik, metode **waco** sebesar 86,08% dengan *delay* pengiriman 2,33 detik, dan metode **waordc** sebesar 79,2% dengan *delay latency* sebesar 3,336 detik seperti pada Tabel 5.7:

Tabel 5.7 Pengaruh Mekanisme Penjadwalan Terhadap Kualitas Jaringan

Method	Energy (mWh)	PDR (%)	Latency (s)
cxmac	169.61	90.88%	4.685
waco	90.16	86.08%	2.330
waordc	68.39	79.20%	3.336

Dari data-data tersebut, diketahui bahwa penggunaan gabungan metode *asynchronous* dan *on-demand waordc* dapat menghemat penggunaan energi dibandingkan metode *asynchronous cxmac* dan metode *on-demand waco* lain. Pengaruh terhadap performa jaringan pada metode **waordc** masih lebih cepat dibandingkan metode *asynchronous cxmac*.



BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Melihat pada hasil percobaan dan analisa yang dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Efisiensi energi WSN menggunakan metode penjadwalan radio pemicu (WuR) dapat dilakukan dengan metode *Wake-up Asynchronous On-demand radio duty-cycle* (**waordc**). Selama pengujian 20 menit, rata-rata kebutuhan energi setiap *node* turun menjadi 68,39 mWh (3,61%), dari total penggunaan energi jika tanpa penjadwalan sebesar 1.890,49 mWh. Hasil tersebut lebih baik dibandingkan metode *on-demand waco* sebesar 90,15 mWh, ataupun mekanisme *asynchronous cxmac* sebesar 169,61 mWh
2. Metode **waordc** memiliki imbal balik pada keberhasilan pengiriman paket. Persentase keberhasilan pengiriman paket (PDR) mengalami penurunan sebesar 79,2% dan menambah *latency* sebesar 3,336 detik.

6.2. Saran

Dari pengalaman percobaan pada penelitian ini, berikut beberapa saran untuk penelitian selanjutnya :

1. Pengaturan waktu *preamble-time* secara dinamis. Metode **waordc** dapat dikembangkan dengan pengaturan *preamble time* secara dinamis berdasarkan pada kepadatan *node* dalam satu area
2. Pengujian langsung pada perangkat keras . Pengujian metode yang kami lakukan hanya sebatas simulasi pada Cooja, masih diperlukan implementasi pada perangkat keras.

DAFTAR PUSTAKA

- Alabi, D. (2017). Energy-Aware Task Scheduling in Contiki.
- Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. *Ad Hoc Networks*, 7(3), 537–568. <https://doi.org/10.1016/j.adhoc.2008.06.003>
- Ansari, J., Pankin, D., & Mähönen, P. (2009). Radio-triggered wake-ups with addressing capabilities for extremely low power sensor network applications. *International Journal of Wireless Information Networks*, 16(3), 118–130. <https://doi.org/10.1007/s10776-009-0100-6>
- Bagula, A., Pietrosemoli, E., Colina, L. A., Vives, A., & Zennaro, M. (2016). *Internet of Things in 5 days*. E-Book. Diambil dari <https://github.com/marcozennaro/IPv6-WSN-book>
- Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006). X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. *Proceedings of the 4th international conference on Embedded networked sensor systems (SenSys 2006)*, 307–320. <https://doi.org/10.1145/1182807.1182838>
- Chukwuka, E., & Arshad, K. (2013). Energy Efficient Mac Protocols for Wireless Sensor Network : a Survey. *International Journal of Scientific & Engineering Research*, 5(4), 75–89.
- Cotrim, J. R., & Kleinschmidt, J. H. (2017). Performance evaluation of RPL on a mobile scenario with different ContikiMAC radio duty cycles. *IEEE International Conference on High Performance Switching and Routing, HPSR, 2017-June*. <https://doi.org/10.1109/HPSR.2017.7968675>
- Decotignie, J., & Hernandez, J. (2005). Low Power MAC Protocols for Infrastructure Wireless Sensor Networks. *Mobile Networks and Application*, 10(5), 675–690.
- Dunkels, a. (2007). Rime-a lightweight layered communication stack for sensor networks. *Eprints.Sics.Se*. Diambil dari <http://dunkels.com/adam/dunkels07rime.pdf>
- El-Hoiydi, A. (2007). Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*, 5, 3418–3423. <https://doi.org/10.1109/ICC.2002.997465>
- Erta Puri Rosidiani, E. L. A. & R. A. (2013). Analisis Variasi Genetik Amorphophallus Muelleri Blume Dari Berbagai Populasi Di Jawa Timur Berdasarkan Sekuen Intron Trnl, 4(6).
- Flores, K. O., Butaslac, I. M., Gonzales, J. E. M., Dumlao, S. M. G., & Reyes, R. S. J. (2017). Precision agriculture monitoring system using wireless sensor network and Raspberry Pi local server. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 3018–3021. <https://doi.org/10.1109/TENCON.2016.7848600>
- Guru, R., Siddaraju, & Anandbabu. (2016). Energy Efficiency Mechanisms in Wireless Sensor Networks: A Survey, *139(14)*, 27–33.



- Jurdak, R., Ruzzelli, A. G., & O'Hare, G. M. P. (2010). Radio sleep mode optimization in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 9(7), 955–968. <https://doi.org/10.1109/TMC.2010.35>
- Kassim, M., Mat, I., & Harun, A. (2014). Wireless Sensor Network in precision agriculture application. *Computer, Information and*, 1–4. Diambil dari http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6878963
- Khajuria, R., & Gupta, S. (2015). Energy optimization and lifetime enhancement techniques in wireless sensor networks: A survey. *International Conference on Computing, Communication & Automation*, 396–402. <https://doi.org/10.1109/CCAA.2015.7148408>
- Kodali, R. K., Rawat, N., & Boppana, L. (2014). WSN Sensors for Precision Agriculture. *2014 IEEE Region 10 Symposium*, (1), 643–648.
- Mat, I., Kassim, M. R. M., & Harun, A. N. (2015). Precision agriculture applications using wireless moisture sensor network. *2015 IEEE 12th Malaysia International Conference on Communications*, (01), 18–23. <https://doi.org/10.1109/MICC.2015.7725400>
- Merlin, C. J., & Heinzelman, W. B. (2010). Duty cycle control for low-power-listening MAC protocols. *IEEE Transactions on Mobile Computing*, 9(11), 1508–1521. <https://doi.org/10.1109/TMC.2010.116>
- Muhammad, A. R. (2018). WSN Based Microclimate Monitoring System on Porang Plantation. *2018 Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, 142–145.
- Noreen, U., Bounceur, A., & Clavier, L. (2017). A study of LoRa low power and wide area network technology. *2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)*, (May), 1–6. <https://doi.org/10.1109/ATSIP.2017.8075570>
- Piyare, R., Istomin, T., & Murphy, A. L. (2017). WaCo : A Wake-Up Radio COOJA Extension for Simulating Ultra Low Power Radios, (February), 48–53.
- Piyare, R., Murphy, A. L., Kiraly, C., Tosato, P., & Brunelli, D. (2017). Ultra Low Power Wake-Up Radios: A Hardware and Networking Survey. *IEEE Communications Surveys and Tutorials*, 19(4), 2117–2157. <https://doi.org/10.1109/COMST.2017.2728092>
- Rajendran, V., Obraczka, K., Garcia-Luna-Aceves, J. J., Tay, Y. C., Jamieson, K., Balakrishnan, H., ... Sinha, P. (2009). Cost- and Collision-Minimizing Forwarding Schemes for Wireless Sensor Networks: Design, Analysis and Experimental Validation. *Wireless Networks*, 22(3), 322–337. <https://doi.org/10.1109/JSAC.2004.830898>
- Rault, T., Bouabdallah, A., & Challal, Y. (2014). Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*. <https://doi.org/10.1016/j.comnet.2014.03.027>
- Sampayo, S. L., Montavont, J., Prégaldiny, F., & Noel, T. (2018). Is Wake-Up Radio the Ultimate Solution to the Latency-Energy Tradeoff in Multi-hop Wireless Sensor Networks? *International Conference on Wireless and Mobile Computing, Networking and Communications, 2018-Octob*. <https://doi.org/10.1109/WiMOB.2018.8589163>

- Schurgers, C., Tsiatsis, V., Ganeriwal, S., & Srivastava, M. (2002). Optimizing Sensor Networks in the Energy-Latency-Density Space. *{IEEE} Transactions on Mobile Computing*, 1(1), 70–80.
- Schurgers, C., Tsiatsis, V., & Srivastava, M. B. (2002). STEM: Topology management for energy efficient sensor networks. *IEEE Aerospace Conference Proceedings*, 3, 1099–1108. <https://doi.org/10.1109/AERO.2002.1035239>
- Tay, Y. C., Jamieson, K., & Balakrishnan, H. (2004). Collision-minimizing CSMA and its applications to wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6), 1048–1057. <https://doi.org/10.1109/JSAC.2004.830898>
- Uwase, M. P., Bezunartea, M., Tiberghien, J., Dricot, J. M., & Steenhaut, K. (2017). Experimental Comparison of Radio Duty Cycling Protocols for Wireless Sensor Networks. *IEEE Sensors Journal*, 17(19), 6474–6482. <https://doi.org/10.1109/JSEN.2017.2738700>
- Vo, T.-T., Nguyen, T.-D., & Vo, M.-T. (2013). Ubiquitous sensor network for development of climate change monitoring system based on solar power supply. *2013 International Conference on Advanced Technologies for Communications (ATC 2013)*, 121–124. <https://doi.org/10.1109/ATC.2013.6698090>
- Wang, L., & Liu, K. (2007). An Adaptive Energy-Efficient and Low-Latency Mac Protocol for Wireless Sensor Networks. *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, 2440–2443. <https://doi.org/10.1109/WICOM.2007.608>
- Xue, Y., & Vaidya, N. F. (2004). A wakeup scheme for sensor networks: achieving balance between energy saving and end-to-end delay. *Real-Time and Embedded Technology and Applications Symposium, 2004. Proceedings. RTAS 2004. 10th IEEE*, 19–26. <https://doi.org/10.1109/RTTAS.2004.1317245>
- Yang, T., Ikeda, M., Mino, G., Barolli, L., Durrresi, A., & Xhafa, F. (2010). Performance evaluation of wireless sensor networks for mobile sink considering consumed energy metric. *24th IEEE International Conference on Advanced Information Networking and Applications Workshops, WAINA 2010*, 245–250. <https://doi.org/10.1109/WAINA.2010.50>
- Ye, W., Heidemann, J., & Estrin, D. (2002). An Energy-Efficient MAC Protocol for Wireless Sensor Networks, *00(c)*, 1567–1576.
- Zhao, Y., & Wu, J. (2010). Stochastic sleep scheduling for large scale wireless sensor networks. *IEEE International Conference on Communications*. <https://doi.org/10.1109/ICC.2010.5502306>

LAMPIRAN

Lampiran 1. Tabel perhitungan kebutuhan energi per-node selama 20 menit.

Jumlah Node per-100x100 m ²	Modul	lama waktu aktif dalam %				konsumsi energi (wH)			
		nullrdc	cxmac	waco	waorcd	nullrdc	cmac	waco	waorcd
3 node	MR	99.9	6.8	0.2	0.2	1878.872	127.652	3.196	3.196
3 node	MR	0.0	0.1	0.0	0.0	0.724	3.982	1.086	1.086
3 node	Mtx	0.0	0.1	0.0	0.0	1.504	3.384	1.128	1.128
3 node	Mrx	0.0	0.0	100.0	67.0	0.000	0.000	64.768	43.416
3 node	WuR	0.0	0.0	0.0	0.0	0.000	0.000	0.216	0.034
3 node	Wtx	0.0	0.0	0.0	0.0	0.000	0.000	0.033	0.011
10 node	MR	99.9	7.2	0.4	0.4	1878.872	135.924	7.708	7.708
10 node	MR	0.0	0.2	0.1	0.1	0.724	7.240	2.172	2.172
10 node	Mtx	0.1	0.1	0.1	0.1	4.512	4.512	1.880	1.880
10 node	Mrx	0.0	0.0	100.0	67.0	0.000	0.000	64.774	43.416
10 node	WuR	0.0	0.0	0.0	0.0	0.000	0.000	0.216	0.034
10 node	Wtx	0.0	0.0	0.0	0.0	0.000	0.000	0.065	0.023
25 node	MR	99.9	9.5	0.9	0.9	1878.872	178.600	15.980	15.980
25 node	MR	0.0	0.5	0.1	0.1	1.086	19.186	2.172	2.172
25 node	Mtx	0.3	0.5	0.1	0.1	11.280	17.296	3.008	3.008
25 node	Mrx	0.0	0.0	100.0	67.0	0.000	0.000	64.768	43.416
25 node	WuR	0.0	0.0	0.0	0.0	0.000	0.000	0.216	0.034
25 node	Wtx	0.0	0.0	0.1	0.0	0.000	0.000	0.229	0.023
50 node	MR	99.9	7.1	2.7	2.7	1878.872	132.916	49.820	49.820
50 node	MR	0.0	0.8	0.1	0.1	1.086	28.960	2.896	2.896
50 node	Mtx	0.7	0.5	0.2	0.2	25.568	18.800	8.648	8.648
50 node	Mrx	0.0	0.0	100.0	67.0	0.000	0.000	64.774	43.416
50 node	WuR	0.0	0.0	0.0	0.0	0.000	0.000	0.216	0.034
50 node	Wtx	0.0	0.0	0.2	0.0	0.000	0.000	0.655	0.023