

Perbaikan Mekanisme *Sleep Scheduling* Secara Dinamis Untuk Jaringan Sensor Nirkabel Berbasis *ZigBee*

Eko Prayitno¹, Waskitho Wibisono²
Program Studi Teknik Informatika, Fakultas Teknologi Informasi,
Institut Teknologi Sepuluh Nopember
Email: eko_prayitno@ymail.com¹, waswib@gmail.com²

ABSTRAK

Penelitian ini mengusulkan perbaikan mekanisme penjadwalan tidur untuk mengurangi konsumsi daya pada JSN, dimana penghematan energi dilakukan dengan cara *scheduling* yang dinamis menggunakan kontrol waktu yang otonom, sehingga dapat *melakukan powering down* pada node selama proses menganggur (*idle*). Penelitian ini menyajikan implementasi secara riil dari node sensor nirkabel berdasarkan sistem arsitektur baru, digunakan untuk mengevaluasi pendekatan yang diusulkan melalui pengukuran konsumsi daya, dan menunjukkan bagaimana mekanisme baru secara adaptif memungkinkan melakukan penghematan energi jika dibandingkan dengan mekanisme NSF. Dari segi keakuratan deteksi kejadian (*event*) *True Positive* untuk mendeteksi gas, mekanisme yang diusulkan memberikan hasil yang lebih baik jika dibandingkan dengan mekanisme *periodic sleep scheduling*, yaitu sebesar 87,5%.

Kata Kunci: *sleep scheduling*, monitoring gas, ketepatan deteksi event, *environment aware*, *ZigBee*, *dynamic sleep scheduling*, Jaringan Sensor Nirkabel

ABSTRACT

We propose a sleep scheduling mechanism on wireless sensor networks, the mechanism aims to reduce power consumption on the wireless sensor network when a node sensor collect data information and immediately transmits the data to the end device nodes. Energy savings are achieved by dynamic synchronization sleep scheduling with autonomous time control, and powering down zigbee nodes during idle times identified through consideration environment-aware for dynamic sleep scheduling mechanism. We present a real implementation on wireless sensor nodes based on a novel architecture system. We evaluate our approach through power consumption measurements, and show how the a new mechanism allows adaptive scheduling for energy savings. In terms of detection accuracy event true value contained gas data derived from the sensors, the proposed mechanism provides better results when compared with periodic sleep scheduling mechanism, that is 87.5%

Keywords: *sleep scheduling*, gas sensor detection, *ZigBee*, *environment aware*, *sleep scheduling*, *wireless sensor network*

1. Pendahuluan

Mengetahui konsumsi energi dan ketepatan dalam mendeteksi data pada Jaringan Sensor Nirkabel (JSN) yang menggunakan teknologi *ZigBee*, sangat berkaitan erat dengan aspek untuk memperpanjang masa penggunaan JSN. Aspek efisiensi untuk memperpanjang masa penggunaan *ZigBee* merupakan suatu hal yang dianggap penting, karena node pada jaringan *ZigBee* memiliki daya

baterai yang terbatas [2]. Efisiensi daya juga sangat dibutuhkan pada aplikasi *ZigBee* yang menggunakan energi *harvest*, hal ini disebabkan karena penerapan energi *harvest* tentunya akan menimbulkan tambahan biaya dan sangat tidak praktis untuk membuat energi tambahan seperti panel tenaga surya pada ruang lingkup yang sangat luas.

Banyak pendekatan yang digunakan untuk menyelesaikan

permasalahan penjadwalan untuk dapat melakukan penghematan energi, salah satunya yaitu mekanisme penjadwalan tidur (*sleep scheduling*) menggunakan mekanisme periodik [1]. Mekanisme *sleep scheduling* secara periodik merupakan suatu mekanisme penjadwalan tidur yang dapat menjalankan proses penjadwalan secara bergantian berdasarkan waktu yang telah ditentukan, seluruh proses pengiriman data dianggap penting dan diberi sejumlah waktu pemrosesan yang disebut kwantra (*quantum*) atau *time-slice* [6]. Mekanisme penjadwalan tidur secara periodik dapat mengurangi waktu idle [19], akan tetapi penerapan mekanisme penjadwalan tidur secara periodik pada JSN dapat menyebabkan beberapa node mengalami kelebihan beban, hal ini disebabkan karena terdapat node dalam keadaan diam (*idle*), sehingga memiliki beban sedikit. Untuk meningkatkan efektivitas distribusi beban kerja terhadap waktu respon, diperlukan adanya mekanisme tambahan yang digunakan sebagai penyeimbang beban [16].

Mekanisme penjadwalan tidur (*sleep-scheduling*) menggunakan mekanisme secara periodik dan adaptif yang telah diteliti sebelumnya, telah terbukti mampu melakukan penghematan energi dengan melakukan pengaturan waktu, akan tetapi pendekatan tersebut hanya mampu melakukan penjadwalan tidur berdasarkan kuantum, namun belum mempertimbangkan aspek pengamatan kondisi lingkungan (*environment aware*). Pada dasarnya terdapat kemungkinan bahwasanya sensing data dapat disesuaikan dengan kebutuhan pengamatan kondisi lingkungan, kapan saatnya suatu *node* melakukan pengambilan keputusan untuk harus mendengarkan suatu informasi dari *node* tetangga dan kapan saatnya *node* tersebut harus melakukan sensing data yang akan dikirim ke-*node coordinator*, sehingga mekanisme tersebut dapat dijadikan pertimbangan untuk menjalankan penjadwalan tidur.

Penelitian yang berusaha melakukan pendekatan dengan cara adaptif untuk penjadwalan tidur yang dinamis dilakukan oleh Hohlt dkk, dimana penjadwalan tidur dapat disesuaikan dengan pengaturan kondisi radio hidup atau mati radio dengan menggunakan mekanisme *sleep scheduling*. Penghematan konsumsi energi dapat dilakukan ketika node dalam keadaan *idle* lalu *sleep*, ketika suatu node dalam kondisi *sleep*, maka radio tersebut berada pada posisi mati, dan kembali hidup untuk melakukan komunikasi data berdasarkan waktu yang telah ditentukan [1].

Dari penjelasan yang telah diuraikan sebelumnya, dalam penelitian ini akan dilakukan perbaikan mekanisme penjadwalan tidur secara adaptif dengan mempertimbangkan aspek pengamatan *environment aware*, dimana pengiriman data tergantung dari pengamatan kondisi lingkungan. Dengan adanya pertimbangan aspek pengamatan *environment aware* diharapkan dapat dapat menghemat konsumsi daya dengan cara mengurangi kecenderungan *node* melakukan pengiriman data secara terus-menerus.

2. Penjadwalan Tidur (*Sleep Scheduling*) JSN

Penjadwalan tidur terbukti telah mampu melakukan penghematan energi dengan baik. Terdapat beberapa cara untuk menjalankan fungsi penjadwalan tidur, diantaranya secara adaptif [6]. Pada dasarnya penjadwalan tidur untuk melakukan penghematan daya dapat dilakukan pengembangan dengan mempertimbangkan kemungkinan pengiriman data dapat disesuaikan dengan tingkat kebutuhan pengiriman dengan kondisi sumber daya yang tersedia, sehingga dapat mempertimbangkan pengambilan keputusan kapan saatnya data harus dikirim ke node tetangga atau data disimpan didalam media penyimpanan sehingga dapat mempertimbangkan kapan node harus menjalankan mekanisme tidur.

2.1 Periodic Sleep Scheduling

Mekanisme tidur periodik dapat menghindari *overhead* pada node sensor, mencegah dari tabrakan serta mengurangi waktu *idle*, hal ini disebabkan karena node sensor secara rutin menjalankan waktu tidur dan bangun secara tetap (statis). Ketika terjadi tebrakan, maka data akan dikirim ulang secara terus menerus sehingga menyebabkan banyak data yang terbuang. Gambar 2.1 memperlihatkan mekanisme sleep scheduling secara periodik, dimana pengaturan waktu telah ditentukan terlebih dahulu.



Gambar 2.1 Periodic Sleep Time

2.2 Flexible Power Scheduling (FPS)

Penghematan daya dengan pendekatan adaptif untuk penjadwalan tidur yang dinamis, sehingga dapat menyesuaikan waktu aktif dengan cara terdistribusi dan mempertimbangkan statistik pengiriman data. Data yang dikirim berselang waktu yang disesuaikan dengan kebutuhan aplikasi, node tidak akan mengirimkan pesan jika terdapat sejumlah data yang sama, dalam kondisi seperti ini node akan diposisikan dalam keadaan *idle*. Dengan melakukan identifikasi penjadwalan yang fleksibel, Penghematan konsumsi energi dapat dilakukan ketika node dalam keadaan *idle*, karena node akan diposisikan dalam keadaan *sleep*. radio akan dimatikan jika tidak mengirim dan menerima pesan, dan akan hidup kembali ketika terdapat komunikasi [1].

Tabel 2.1 memperlihatkan komunikasi dengan menggunakan mekanisme sleep scheduling antara dua node [1]. Cycle (C) merupakan jumlah siklus yang terjadi sedangkan Slot (S) merupakan jumlah slot. Komunikasi dimulai dengan Node 1 (parent) mengirimkan advertise dan diterima oleh node 2, selanjutnya node 2 merespon dan

node 1 segera mengirimkan konfirmasi, dan langkah yang ketiga merupakan pengiriman data sensor.

Tabel 2.1 Mekanisme komunikasi FPS

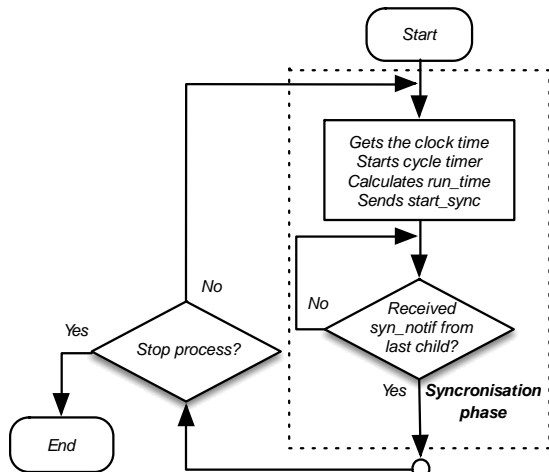
| C | S | Node 1 | Node2 |
|---|---|-----------------------------|-----------------------|
| 1 | A | Advertise slot R | Receive advertisement |
| 2 | R | Send immediate confirmation | Request slot R |
| 3 | R | Receive from Node 2 | Transmit to Node 1 |
| . | . | ... | ... |
| N | R | Receive from Node 2 | Transmit to Node 1 |

2.3 Dynamic Sleep scheduling

Pendekatan menggunakan mekanisme *dynamic sleep scheduling* menempatkan satu *node* dalam keadaan aktif, sedangkan *node* lainnya dikondisikan dalam keadaan *sleep*. Dengan pendekatan ini hanya *node* aktif yang akan menghabiskan banyak energi, dan sebaliknya *node* lainnya dalam keadaan tidur, sehingga dapat melakukan penghematan energi guna memperpanjang penggunaan node tersebut. Mekanisme *dynamic sleep scheduling* merupakan suatu mekanisme untuk melakukan pengaturan pada *node*, kapan saatnya *node* harus aktif dan *node* lainnya dalam keadaan *sleep*. selama *node* dalam keadaan aktif maka *node* tersebut akan mengirimkan data yang berasal dari sensor yang ditujukan ke *base station* [15].

3. Time Synchronitation

Gambar 3.2 menunjukkan aliran sistem pada tahap sinkronisasi waktu yang terjadi pada *gateway*. Pada tahap sinkronisasi, *gateway* memperoleh sejumlah waktu berdasarkan jam untuk melakukan sinkronisasi. pengaturan *sleepscheduling* dalam kondisi aktif atau *sleep* berdasarkan waktu yang telah ditentukan. mengirim atau menerima pesan dari node lain berdasarkan batas waktu yang telah ditentukan [4].



Gambar 3.2 Aliran Sistem Sinkronisasi Waktu (Azevedo dkk., 2014)

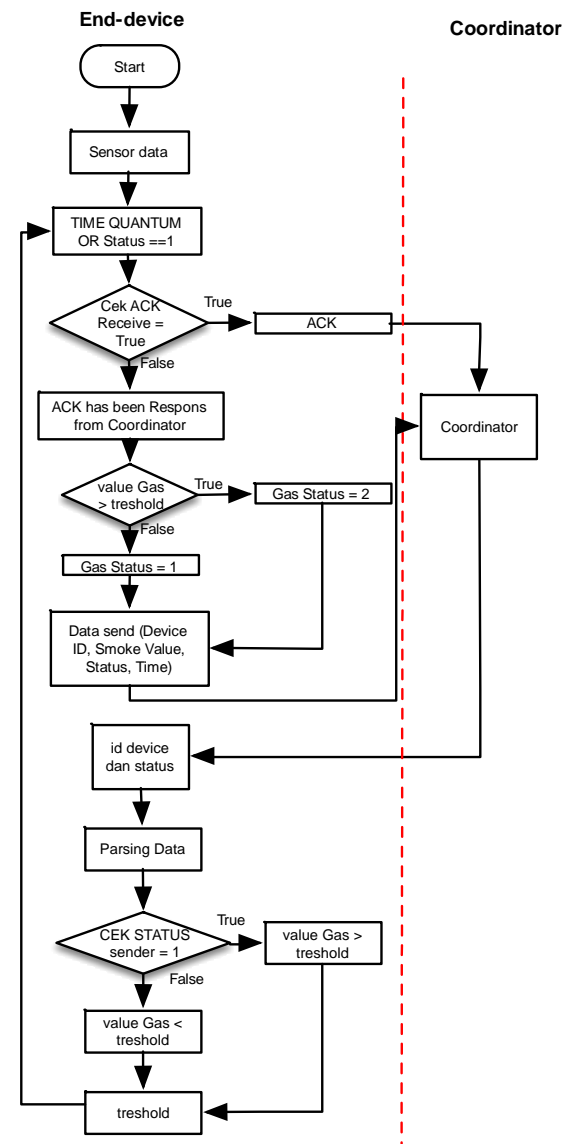
4. Metodologi Penelitian

4.1. Mekanisme yang Diusulkan

Desain algoritma pada penelitian ini meliputi desain algoritma sistem secara keseluruhan seperti yang ditunjukkan pada Gambar 3.6. dan gambar 3.7 Algoritma sistem terdiri dari dua, yaitu *transfer (Tx)* dan *receive (Rx)* yang dilakukan oleh *end-devicenode* (Gambar 3.6) dan *coordinator node* (Gambar 3.7). Setiap *node* dilengkapi dengan sensor, yang bertujuan untuk menerima data *environment* yang diperoleh dari sekitar area uji coba. data tersebut nantinya akan diteruskan ke *nodecoordinator*. Pengiriman data dilakukan berdasarkan penjadwalan waktu yang telah ditetapkan, pengiriman data secara langsung terjadi karena adanya *interrupt* dari sensor yang mengirimkan data diatas ambang (*threshold*) proses tersebut terjadi perangkat pada *end devive*.

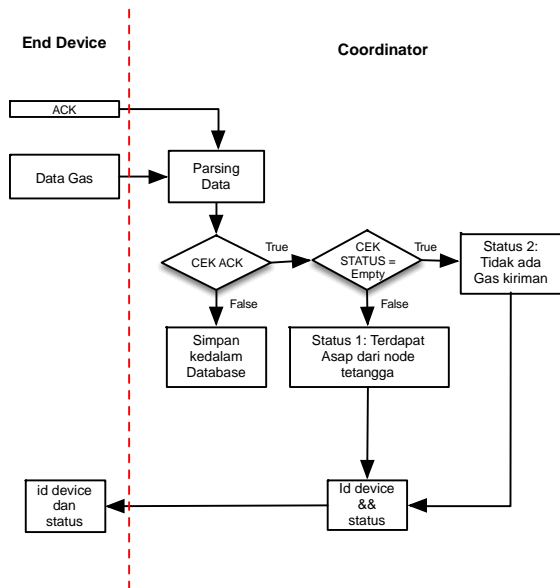
Jika tidak terjadi *interrupt* yangdiakibatkan oleh ambang *threshold* maka memungkinkan pengiriman data dapat dilakukan secara bersama-sama dengan *node* tetangga, dalam artian pengiriman data dapat dilakukan lebih dari 1 *node*, sehingga dengan kondisi speerti ini, pengiriman data dapat dijadwalkan secara adaptif berdasarkan kondisi sensing lingkungan, jika sensing kondisi lingkungan terdapat indikasi meningkat,

maka memungkinkan sensor yang aktif memiliki waktu sinkronisasi yang lebih lama, sebaliknya jika kondisi lingkungan didapati indikasi lingkungan sensing menurun maka sensor yang aktif akan lebih sedikit.



Gambar 4.1. Diagram Metode Penelitian

Gambar 3.10 menggambarkan penjelasan dengan menggunakan diagram alir, bagaimana mekanisme *sleep* denganmempertimbangkan kondisi lingkungan sehingga pemberian nilai *threshol* untuk menentukan adaptif time dapat dilakukan.



Gambar 4.2 Diagram Alir Coordinator Node

Tabel 2.2 Pseudocode Mengirim Data

```

    Create thread sending data
    Set pointer thread sending
    Set interval thread
    Set time
    if ((time < time quantum) or (Gas value > threshold)) then
        XBee wake Up
        if (Receive acknowledgement == false) then
            Sending data <-- id, ack, time
        else
            Sending ACK
        if (Temporary Gas Condition ≠ Gas Condition) then
            Sending data <--id, Gas value, Gas condition , time
            Temporary status = Gas Condition
        else
            ACK receive= false
        else
            Xbee Go to sleep
    
```

Tabel 2.3 Pseudocode Menerima Data

```

    Create thread receive data
    Set pointer thread receive
    Set interval thread
    while (data available) then
        read character
        if (character = #) then
    
```

```

    parsing data= true
    acknowledgment = true
    break
    if (parsing data) then
        set function parsing data
        parsing data = false
        clear data in
    
```

Tabel 2.4 Pseudocode Parsing Data

```

    n=0
    n = [data in]
    for ( i=0; i<length data in; i++) then
        if (data in [i] = ',' or ( data in [i] = '#')) then
            n++
            set data [n] = ""
        else
            data [n] = data [n] + data [i]
        else
            Display and sending parsing data
    
```

Tabel 2.5 Pseudocode Penjadwalan waktu Sleep

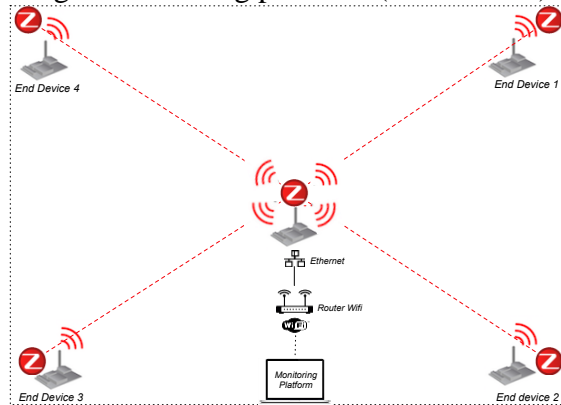
```

    Set time
    if ((time < time quantum) or (Gas value > threshold)) then
        Wake Up
        Sending data <-- id, ack, time
    else
        Sending ACK
        if (Temporary Gas Condition ≠ Gas Condition) then
            sending data <--id, Gas value, Gas condition , time
        else
            ACK receive= false
    else
        Go to sleep
    
```

4.2. Perancangan Uji Coba

Arsitektur umum yang akan dibangun dalam penelitian ini adalah terdiri dari *node end device* dan *node coordinator*. Gambar 3.2 merupakan rancangan arsitektur umum sistem, yang terdiri dari *node end device* dan

coordinator. Komunikasi antara *node end device* dan *coordinator* dilakukan oleh perangkat *Xbee*. Untuk dapat proses monitoring data, maka diperlukan penghubung antara monitoring platform dengan perangkat *coordinator*, maka pada penelitian ini digunakan router wifi sebagai penghubung antara perangkat *coordinator* dengan monitoring platform (Gambar 3.2).



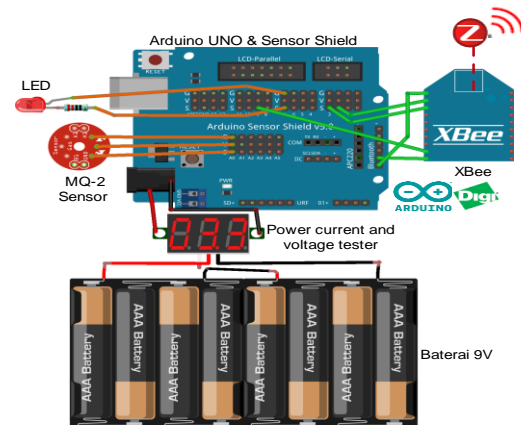
Gambar 4.3 Arsitektur Umum Sistem

4.2.1 End-device Node

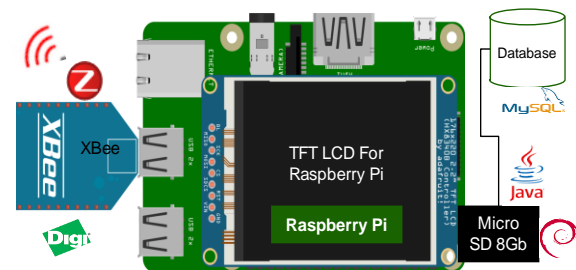
Node *end device* bersifat homogen yang memiliki modul dan kemampuan yang sama serta bersifat statik atau tidak bergerak. Sensor yang digunakan untuk melakukan sensing kondisi *environment* dipasang pada perangkat *end device*. Adapun modul yang digunakan *node end device* ditunjukkan pada gambar 3.2 rancangan arsitektur *node end device*. Kemampuan yang dimiliki *node end device* antara lain, melakukan aktifitas *sensing*, mendengarkan *Acknowledgement (ACK)*, mengirim dan menerima paket.

4.2.2 Coordinator Node

Guna mempermudah membuat analisa, pada penelitian ini, maka *node coordinator* hanya menggunakan 1 buah *node*. *node coordinator* menggunakan *mini PC raspberry* yang dilengkapi dengan *TFT LED touch screen* serta *XBee* untuk komunikasi dengan *node end device*. *Nodecoordinator* berfungsi untuk menerima data dari *node end device*, dan selanjutnya data tersebut akan disimpan kedalam *database*.



Gambar 4.4 Rancangan Arsitektur *end device node*

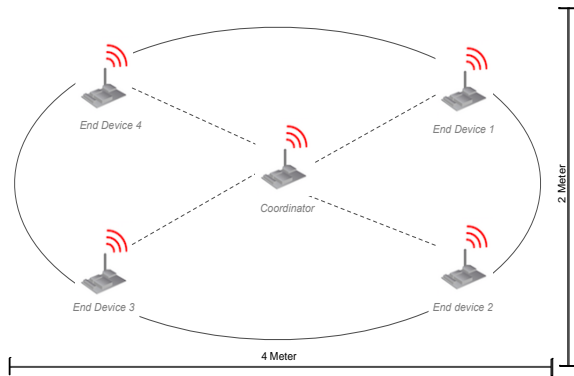


Gambar 4.5 Rancangan Arsitektur *Coordinator Node*

4.2.3 Lingkungan Uji Coba

Lokasi uji coba dilakukan didalam ruangan, pemilihan lokasi ujian tersebut didasarkan pada ruangan tertutup yang memiliki angin lebih sedikit, hal ini ditujukan agar mempermudah melakukan analisa data berdasarkan skenario yang telah dirancang.

Uji coba dilakukan dengan menggunakan lima *node*, yang terdiri dari empat *node end-device* dan satu *node coordinator*. Setiap *node end device* disebarakan pada sudut ruangan yang berbentuk persegi panjang, sedangkan *coordinator* diposisikan tepat ditengah-tengah ruangan yang berbentuk persegi panjang (Gambar 4.1). Spesifikasi perangkat dan lingkungan uji coba lingkungan diperlihatkan pada tabel 4.1.



Gambar 4.6 Lingkungan Uji Coba

Tabel 4.1 Spesifikasi Lingkungan Uji Coba

| Lingkungan Uji Coba | Keterangan |
|----------------------------------|---|
| Ukuran Ruang | 2 x 4 Meter |
| Jumlah Node | 5 Node |
| Topologi | Star |
| End-device Node | Mikrokontroler Arduino, sensor gas MQ-2, sensor shield v1.1, lampu LED, XBee, sumber daya (resource) |
| Rata-rata energi awal end-device | 170 mA |
| Coordinator Node | Raspberry Pi 2 yang dilengkapi dengan LCD TFT monitor 3,3" dengan OS Raspbian, Xbee dan modul gateway |
| Monitoring | Laptop, Router beserta modem |
| Event detection | Gas |

4.3. Pengujian

Uji coba ini dilakukan untuk mengamati dan menganalisa mekanisme yang diusulkan, untuk mempermudah melakukan hal tersebut, maka dibutuhkan mekanisme pembanding. Mekanisme pembanding yang digunakan pada penelitian ini adalah mekanisme *naïve store and forward* [1] dan mekanisme *periodic sleep scheduling*.

4.3.1 Parameter Pengujian

Beberapa parameter pengujian yang digunakan pada penelitian ini adalah: Penggunaan energi dan perhitungan *event detection* mekanisme *naïve store and forward*.

Penggunaan energi dan perhitungan *event detection* mekanisme periodik. Penggunaan energi dan perhitungan *event detection* mekanisme yang diusulkan

4.3.2 Skenario Pengujian

Skenario pengujian dilakukan untuk mengetahui penggunaan arus dan tegangan pada masing-masing *node end-device*. Pengamatan penggunaan arus pada masing-masing *end-device* bertujuan untuk mengetahui estimasi penggunaan daya setiap mekanisme yang akan dilakukan pengujian, adapun skenario pengujian yang dilakukan, adalah sebagai berikut:

4.3.2.1 Uji Coba Menggunakan Mekanisme *Naïve Store and Forward*

Mekanisme *naïve store and forward (NSF)* dirujuk dari penelitian yang dilakukan oleh (Hohlt., dkk 2004). Pengiriman data pada mekanisme NSF dilakukan secara terus menerus, baik itu ketika kondisi terjadi kejadian gas ataupun tidak ada kejadian gas (kondisi *idle*). Skenario pengujian dilakukan dengan menggunakan 1 *node coordinator* dan 4 *node end device sesuai dengan* lingkungan ujicoba. Dalam penelitian ini, untuk mengamati arus dan tegangan yang terjadi pada *node end device*, maka dibutuhkan perangkat tambahan, yaitu volt meter dan arus *detector*. Pengujian dilakukan sesuai dengan skenario sub bab pelaksanaan uji coba.

4.3.2.2 Uji Coba Menggunakan Mekanisme *Periodic Sleep Scheduling*

Mekanisme *periodic sleep scheduling* merupakan mekanisme yang mengatur

setiap *end device* dalam keadaan hidup atau mati. Pengaturan tersebut dilakukan secara *autonomous* oleh *perangkat end device*. *End device* akan dimatikan sesuai dengan waktu yang telah ditentukan, dan hidup kembali juga berdasarkan waktu yang telah ditentukan. Uji coba mekanisme *periodic sleep scheduling* dilakukan berdasarkan lingkungan uji coba dan pelaksanaan uji coba yang telah dijelaskan pada sub bab sebelumnya.

Dalam percobaan ini dilakukan pengukuran arus dan tegangan pada *node end device* dalam kondisi *wake-up* dan sensor gas MQ-2 dalam kondisi aktif melakukan sensing, dan selanjutnya meneruskan paket data tersebut ke *node coordinator*. Pengukuran dilakukan terhadap kedua mekanisme *sleep scheduling* dalam penelitian ini, yaitu dalam keadaan aktif mengirimkan paket dari *node end device* ke *node coordinator*. konsumsi energi dilakukan pengukuran terhadap seluruh *node end device* selama *cycle time* yang telah ditentukan.

4.3.2.3 Uji Coba Menggunakan Mekanisme yang Diusulkan

Tujuan utama dari uji coba *sleep scheduling* adalah untuk mengurangi konsumsi daya. *Sleep scheduling* dilakukan secara *autonomous* oleh *node end device* berdasarkan waktu (*t*) yang telah ditentukan. *Sleep Scheduling* beradaptasi secara dinamis berdasarkan pengamatan kondisi lingkungan (*environment-aware*) sebagai perubahan permintaan penjadwalan. pada percobaan ini dengan kondisi *node* dalam kondisi *sleep* maka *node end device* tersebut tidak dapat melakukan pengiriman atau menerima data. Uji coba dilakukan sebanyak 15 kali *cycle time* pada saat *node* dalam keadaan *sleep*.

Pengukuran dilakukan dengan manajemen daya diaktifkan dan satu pengukuran diambil dengan manajemen daya dinonaktifkan. Ketika manajemen daya diaktifkan, uji coba dilakukan

dengan kondisi radio dalam keadaan *sleep* selama waktu yang telah ditentukan. Ketika manajemen daya dinonaktifkan merupakan kebalikan dari kondisi aktif dimana uji coba dilakukan pada saat radio saling berkomunikasi antara *node end device* dan *coordinator*. Konsumsi energi dilakukan pengukuran tegangan dan arus terhadap seluruh *node end device*.

5. Hasil dan Pembahasan

Hasil uji coba dan analisa dilakukan terhadap mekanisme *naïve forward and store*, *periodic* dan mekanisme yang diusulkan.

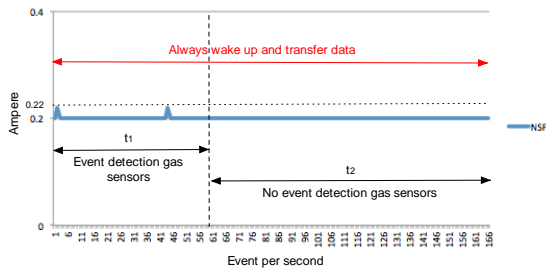
5.1 Hasil Uji Coba Menggunakan Mekanisme *Naïve Forward and Store*

Pengujian penggunaan daya baterai dilakukan untuk mengetahui berapa jumlah penggunaan arus dan tegangan pada masing-masing *end device*, dengan melakukan perbandingan terhadap mekanisme yang diusulkan yaitu *improvement Dynamic environment aware sleep scheduling* dengan mempertimbangkan aspek pengamatan kondisi lingkungan yang akan dibandingkan dengan mekanisme *naïve store and forward* dan mekanisme *periodik*. Pengamatan uji coba untuk mengetahui penggunaan daya baterai dilakukan sebanyak 15 kali *cycle time*.

Hasil uji coba menggunakan *naïve store dan forward* menunjukkan bahwasanya penggunaan arus relatif sama pada kondisi terdapat kejadian untuk mendeteksi gas ataupun tidak terdapat kejadian untuk mendeteksi gas. Penggunaan arus relatif sama yaitu 0,2 *Ampere* (A) atau 20 *mili ampere* (mA) baik dalam keadaan terdapat kejadian mendeteksi gas ataupun tidak, hal ini disebabkan karena *node end device* tidak dilakukan penjadwalan tidur, sehingga *node end device* selalu mengirimkan pesan walaupun tidak terdapat kejadian

mendeteksi gas. Jumlah kejadian yang dilakuakn pengamatan dalam uji coba ini sebanyak 240 kejadian, dimana pengambilan data untuk setiap kejadian dilakukan perdetik (Gambar 4.3).

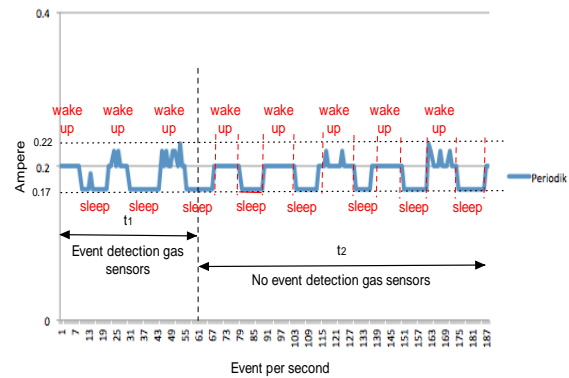
Total penggunaan arus untuk menjalankan mekanisme *naïve store and forward* adalah sebesar 20.023 mA



5.1 Grafik hasil uji coba menggunakan mekanisme *naïve store and forward*

5.2 Hasil Uji Coba Menggunakan Mekanisme *Periodic Sleep Scheduling*

Perbandingan rata-rata penggunaan arus dan tegangan untuk kedua mekanisme periodik menunjukkan bahwasanya mekanisme *periodic sleep scheduling* memberikan hasil yang relatif berbeda untuk setiap *end device*. dengan menggunakan mekanisme periodik *sleep scheduling* mampu menurunkan beban penggunaan arus jika dibandingkan dengan mekanisme NFS, hal ini deisbabkan karena mekanisme periodik *sleep scheduling* diberikan sejumlah waktu, kapan saatnya node harus bangun dan kapan saatnya node dijadwalkan tidur selama 15 detik. Mekanisme *periodic sleep scheduling* tidak akan mendeteksi gas walaupun ketika terdapat gas, ha ini disebabkan karena wkatu yang dijadwalkan untuk bangun telah habis.



5.2 Mekanisme Periodik *Sleep Scheduling*

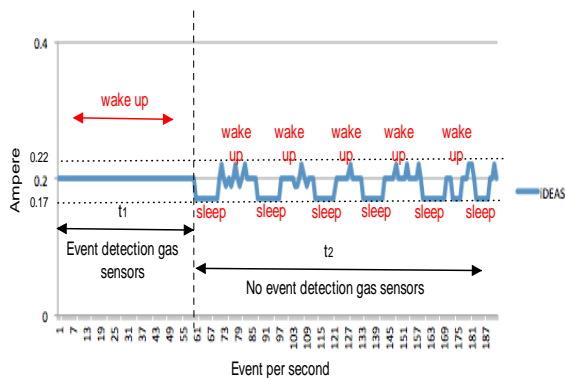
Total Penggunaan arus untuk menjalankan mekanisme *periodic sleep scheduling* adalah sebesar 18.702 mA, jika dibandingkan dengan mekanisme NFS tentunya mekanisme periodik mampu melakukan penghematan daya, namun menggunakan mekanisme *periodic sleep scheduling* memiliki kelemahan untuk mendeteksi gas ketika system tidak dijadwalkan untuk bangun.

5.3 Hasil Uji Coba Menggunakan Mekanisme yang diusulkan

Hasil Uji Coba menunjukkan bahwasanya mekanisme yang diusulkan mampu menjalankan mekanisme adaptif. Beda halnya dengan mekanisme periodik, dimana mekanisme periodik tidak dapat mendeteksi gas ketika waktu yang dijadwalkan telah habis, pada mekanisme yang diusulkan ini ketika terdapat gas namun waktu tidak dijadwalkan maka sistem akan mampu beradaptasi utnuk memperpanjang waktu sensing. Gambar 4.5 menunukan system beradaptasi pada menit pertama selanjutnya pada kejadian berikutnya ketika tidak terdapat gas maka mekanisme *sleep scheduling* kembali susai dengan penjadwalan yang telah ditentukan.

Total penggunaan arus untuk mekanisme yang diusulkan untuk menjalankna sesuai skenario yang telah diusulkan adalah sebesar 19.212 mA.

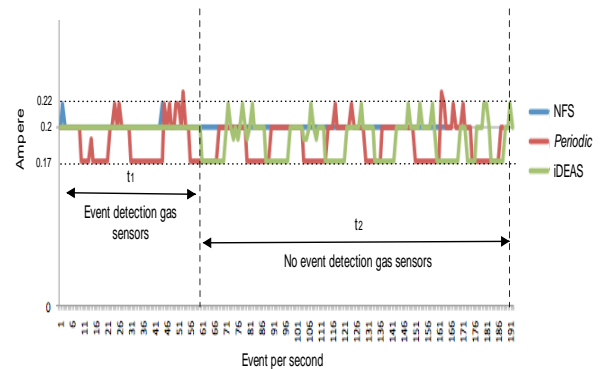
Total penggunaan arus mekanisme yang diusulkan lebih besar daripada mekanisme *periodic sleep scheduling*, hal ini disebabkan karena mekanisme ini memperpanjang waktu untuk beradaptasi berdasarkan pengamatan lingkungan, sehingga ketika memperpanjang waktu akan mengkonsumsi sejumlah arus. Namun dari segi mendeteksi kejadian, mekanisme ini memiliki kelebihan, yaitu mampu mendeteksi walupun dalam keadaan tidak dijadwalkan dan memperpanjang waktu mendeteksi selama terdapat *event* kejadian gas.



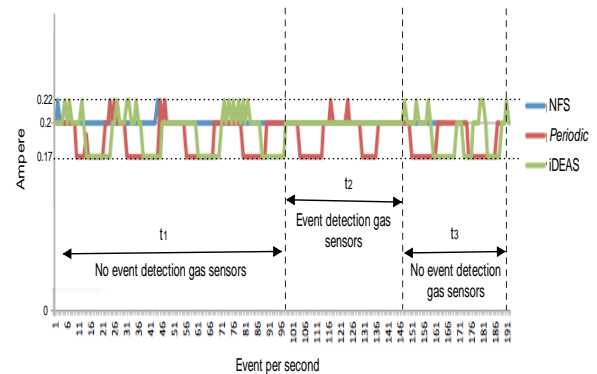
Gambar 5.3 Mekanisme yang diusulkan

Gambar 5.4 perbandingan antara mekanisme yang diusulkan memperlihatkan dimana penggunaan arus mekanisme yang diusulkan lebih besar daripada mekanisme *periodic sleep scheduling*, hal ini disebabkan karena mekanisme ini memperpanjang waktu untuk beradaptasi berdasarkan pengamatan lingkungan, sehingga ketika memperpanjang waktu akan mengkonsumsi sejumlah arus. Namun dari segi mendeteksi kejadian, mekanisme ini memiliki kelebihan, yaitu mampu mendeteksi walupun dalam keadaan tidak dijadwalkan dan memperpanjang waktu mendeteksi selama terdapat kejadian gas. Jika dibandingkan dengan mekanisme NFS mekanisme yang diusulkan lebih unggul dalam hal penggunaan arus. Pola penggunaan arus pada setiap *end device*

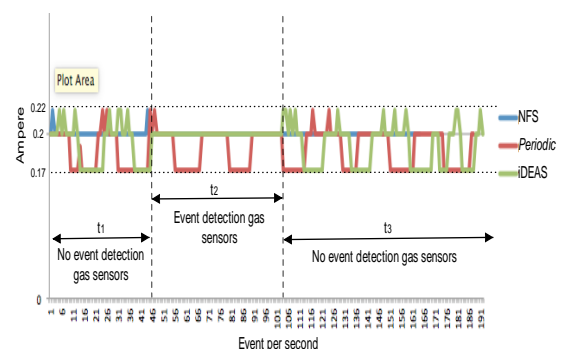
relative sama, yang ditunjukkan pada gambar *sleep5.5*, *5.6* dan *5.7*



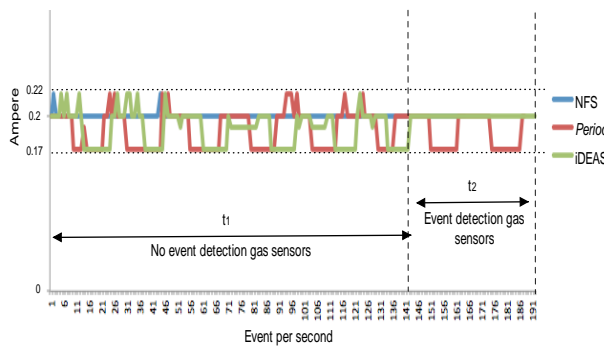
Gambar 5.4 Perbandingan rata-rata Penggunaan Arus pada setiap mekanisme Pengujian End Dvice 1



Gambar 5.5 Perbandingan rata-rata Penggunaan Arus pada setiap mekanisme Pengujian End Dvice 3

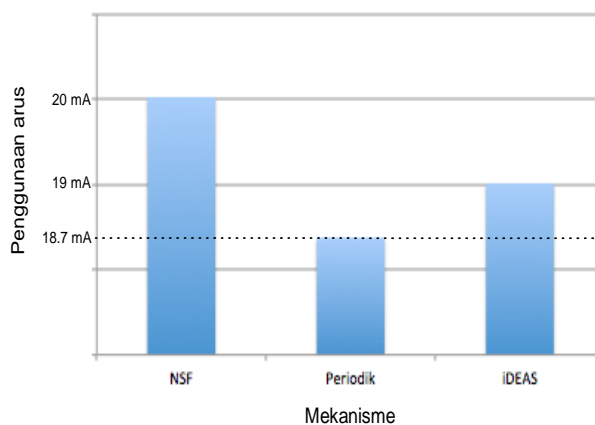


Gambar 5.6 Perbandingan rata-rata Penggunaan Arus pada setiap mekanisme Pengujian End Dvice 2



Gambar 5.7 Perbandingan rata-rata Penggunaan Arus pada setiap mekanisme Pengujian End Device 4

Rata-rata penggunaan arus dan tegangan setiap end device dengan kondisi sleep, dengan menggunakan mekanisme yang diusulkan, telah berhasil menurunkan beban penggunaan arus jika dibandingkan dengan mekanisme NFS, namun ketika dibandingkan dengan mekanisme periodik pengamatan dilakukan sebanyak 15 *cycle time* dengan menggunakan mekanisme *dynamic sleep scheduling* sebagai mekanisme yang diusulkan dan dibandingkan dengan mekanisme *naive store and forward*.



Gambar 5.8 Grafik Perbandingan Mekanisme Uji Coba

Dari perbandingan dua skenario pengujian mekanisme *sleep scheduling* yang diusulkan dan *Naive Store and forward* disimpulkan bahwa rata-rata penggunaan arus pada node *end device*

yang menggunakan mekanisme *dynamic sleep scheduling* lebih baik, yakni terjadi penurunan arus sebesar 0.1A atay 100mA. Hal ini disebabkan karena mekanisme *dynamic sleep scheduling* berada pada kondisi sleep sesuai waktu yang telah ditentukan, sehingga dampak positif dari penggunaan mekanisme *dynamic sleep scheduling* ini lebih sedikit mengkonsumsi arus. Namun jika dibandingkan dengan mekanisme *sleep scheduling* secara periodik, mekanisme yang diusulkan lebih banyak mengkonsumsi arus, hal ini disebabkan mekanisme yang diusulkan beradaptasi sesuai dengan pengamatan lingkungan sehingga memperpanjang waktu sensing. Oleh sebab itu pada pengujian berikutnya akan dilakuakn pengujian perhitungan ketepatan deteksi kejadian pada pengiriman data sensor ke *coordinator*, hal ini dilakukan untuk mengamati dan menganalisa ketepatan deteksi event setiap mekanisme yang diusulkan.

6. Kesimpulan

Berdasarkan hasil dari ujicoba yang dilakukan, diketahui bahwa penggunaan arus pada mekanisme yang diusulkan lebih besar daripada mekanisme *periodic sleep scheduling*, hal ini disebabkan karena mekanisme ini memperpanjang waktu untuk beradaptasi berdasarkan pengamatan lingkungan, sehingga mengkonsumsi sejumlah arus. Namun dari segi deteksi kejadian untuk melakukan sensing data, mekanisme yang diusulkan memiliki kelebihan, yaitu mampu mendeteksi kejadian walupun dalam keadaan tidak dijadwalkan dan mampu beradaptasi untuk memperpanjang waktu deteksi selama terdapat event.

Berdasarkan hasil ujicoba, jumlah penggunaan arus dengan menggunakan mekanisme yang diusulkan jika dibandingkan mekanisme *naif store andforward*, mekanisme yang diusulkan

mampu menghemat penggunaan arus, hal ini disebabkan karena mekanisme NFS tidak memiliki penjadwalan tidur, sehingga menamabah beban penggunaan arus.

Dari Hasil Uji Coba yang dilakukan terhadap perubahan *event* mekanisme yang diusulkan dalam hal mendeteksi tingkat ketepatan (*precision*) untuk mendeteksi gas adalah sebesar 87,5%,

7. Saran

Berdasarkan hasil uji coba yang telah dilakukan, untuk dapat meningkatkan hasil dari mekanisme yang diusulkan, dapat dilakukan pemilihan algoritma sleep scheduling yang lebih baik agar model energi yang digunakan dapat diperbaiki sesuai dengan skenario percobaan mekanisme yang diusulkan.

Hasil pengujian menunjukkan bahwasanya peningkatan yang signifikan akan didapatkan apabila dilakukan *sleep scheduling* secara periodik akan tetapi *periodic sleep scheduling* tanpa mempertimbangkan aspek adaptif akan mempengaruhi terhadap ketepatan deteksi kejadian data sensor, sehingga dibutuhkan suatu algoritma untuk mengatasi pengambilan keputusan mekanisme *sleep scheduling*.

Daftar Pustaka

- [1] B. Hohlt, et al., (2004), "Flexible Power Scheduling for Sensor Networks". In Information processing in sensor networks, Hal 205-214, ACM
- [2] Guofang Nan, et al., (2012), "CDSWS: Coverage-Guaranteed Distributed Sleep/Wake Scheduling for Wireless Sensor

Networks", Springer International Publishing AG Hal. 1687-1499, Springer.

- [3] Guanxiong Shi. Et al., (2011), "Comprehensive Review of Sleep/Wake Scheduling in Wireless Sensor Networks" Communications in Computer and Information Science Vol. 163. Hal. 492-499, Springer.
- [4] Azevedo, J. et al., (2014), "Sleeping ZigBee networks at the application layer". In Wireless Sensor Systems, (IET) Vol. 4. Hal. 35-41, IEEE.
- [5] Ghildiyal, S. et al., (2014), "An Overview of Wireless Sensor Networks Using Zigbee Technology". Computer science and software engineering Vol.2, Available online at: www.ijarcse.com
- [6] J. Elson and D. Estrin, (2001) "Time Synchronization for Wireless Sensor Networks". International Parallel and Distributed Processing Symposium (IPDPS) Available online at: www.isi.edu/scadds/papers/timesync.pdf
- [7] Wen-Tsai Sung et al., (2010) "Multi-Sensors Data fusion for Precise Measurement based on ZigBee

- WSN via Fuzzy Control” In Computer Communication Control and Automation (3CA), Hal 156 – 159, IEEE
- [8] Vyas, A et al., (2014)“Fuzzy Based Sleep Scheduling in TDM Passive Optical Networ“, Communication Systems and Network Technologies (CSNT), Hal. 524 – 528, IEEE
- [9] Elemenreich W. (2002)“*Sensor fusion time-triggered system PhD thesis*” Institute fur Technische Informatik, Technische Universitate Wien
- [10] Liu Weili, et al., (2011)“Detection of fire based on multi-sensor fusion” International Conference on Computer Science and Network Technology (ICCSNT), Hal. 223 – 227, IEEE
- [11] Azharuddin, M., Kuila, P., & Jana, P. K. (2013). “A distributed fault-tolerant clustering algorithm for wireless sensor networks”. In International Conference on Advances in Computing, Communications and Informatics (ICACCI), hal. 997-1002. IEEE.
- [12] Anind K. Dey dan Gregory D. Abowd (2000) “Towards a Better Understanding of Context and Context-Awareness”, Proceedings of the CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands, <ftp://ftp.cc.gatech.edu/pub/gvu/tr/1999/99-22.pdf>
- [13] Mari Korkea-aho, “*Context-Aware Applications Survey*”, <http://www.cse.tkk.fi/fi/opinnot/T-110.5190/2000/applications/context-aware.html>
- [14] Huadong Wu (2003), “*Sensor Data Fusion for Context-Aware Computing Using Dempster-Shafer Theory in partial fulfillment of the Requirements for the degree of Doctor of Philosophy in Robotics*”, https://www.ri.cmu.edu/pub_files/pub4/wu_huadong_2003_1/wu_huadong_2003_1.pdf
- [15] Michal’ik, M. (2013). “*Base station for Wireless sensor network*”. Masaryk University. Brno, Czech Republic.
- [16] Miksatko, Jan (2004) “*Dynamic load balancing of fine-grain services using prediction based on service input*”. A THESIS Submitted in partial fulfillment

of the requirements for the degree master of science, kansas university

- [17] Jayant Gupchup, dkk (2009), *Model-Based Event Detection in Wireless Sensor Networks*
- [18] Pawar, 2011 “Life Time Prediction of Battery Operated Node for Energy Efficient WSN Applications”.
<http://www.ijcst.com/vol24/3/pawar.pdf>
- [19] Krisda Khankasikam, (2013) “An Adaptive Round Robin Scheduling Algorithm: A Dynamic Time Quantum Approach”, In International Journal of Advancements in Computing Technology (IJACT) Vol. 5, no.1,
<http://www.aicit.org/IJACT/ppl/IJACT1958PPL.pdf>