

**IMPLEMENTASI JARINGAN ZIGBEE  
PADA SISTEM *SMART HOME* BERBASIS SIMULASI  
MENGUNAKAN OMNET++**



Oleh:

Mochammad Bachrul Ulum (225150200111022)

Tasya Kamila Rahmasari (225150200111024)

Maritza Aliyya Devy (225150200111026)

**UNIVERSITAS BRAWIJAYA  
FAKULTAS ILMU KOMPUTER  
MALANG  
2024**

## DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>2</b>
<b>DAFTAR GAMBAR.....</b>	<b>3</b>
<b>DAFTAR TABEL.....</b>	<b>4</b>
<b>BAB I.....</b>	<b>5</b>
<b>PENDAHULUAN.....</b>	<b>5</b>
1.1. Latar Belakang.....	5
1.2. Rumusan Masalah.....	6
1.3. Tujuan.....	6
1.4. Manfaat Penelitian.....	6
1.5. Batasan Masalah.....	6
1.6. Sistematika Pembahasan.....	7
<b>BAB II.....</b>	<b>8</b>
<b>TINJAUAN PUSTAKA.....</b>	<b>8</b>
2.1. Smart Home.....	8
2.2. OMNeT++.....	8
2.3. Zigbee.....	9
2.4. IEEE 802.15.4.....	10
2.5. INET Framework.....	11
<b>BAB III.....</b>	<b>13</b>
<b>METODOLOGI PENELITIAN.....</b>	<b>13</b>
3.1. Tipe Penelitian.....	13
3.2. Strategi dan Rancangan Penelitian.....	13
3.2.1. Strategi Penelitian.....	13
3.2.2. Subjek Penelitian.....	13
3.2.3. Lokasi Penelitian.....	13
3.2.4. Metode Pengumpulan Data.....	13
3.2.5. Metode Analisis Data.....	13
3.2.6. Peralatan Pendukung.....	13
3.3. Perancangan.....	14
3.3.1. Arsitektur Sistem.....	14
3.3.1.1. Network Layer.....	14
3.3.1.2. Application Layer.....	14
3.3.1.3. Komponen Simulasi.....	14
3.3.2. Framework.....	14
3.3.3. Topologi Jaringan.....	15
3.4. Tahap Implementasi.....	16
3.4.1. Implementasi Network Layer.....	16
3.4.3. Implementasi Smart Lamp.....	17

3.4.4. Implementasi Smart Curtain Opener.....	17
<b>BAB IV.....</b>	<b>18</b>
<b>HASIL DAN PEMBAHASAN.....</b>	<b>18</b>
4.1. Hasil.....	18
<b>BAB V.....</b>	<b>26</b>
<b>PEMBAHASAN.....</b>	<b>26</b>
5.1. Pembahasan.....	26
<b>BAB VI.....</b>	<b>27</b>
<b>PENUTUP.....</b>	<b>27</b>
6.1. Kesimpulan.....	27
6.2. Saran.....	27
<b>DAFTAR PUSTAKA.....</b>	<b>28</b>

## DAFTAR GAMBAR

Gambar 3.1. Desain Topologi Star.....	16
Gambar 3.2. Tampilan Desain Implementasi Topologi Dalam OMNeT++.....	17
Gambar 4.1. Flowchart Skenario 1.....	21
Gambar 4.2. Flowchart Skenario 2.....	23

## DAFTAR TABEL

Tabel 4.1. Response Time (ms).....	24
Tabel 4.2. Tingkat Keberhasilan (%).....	25
Tabel 4.3. Konsumsi Daya (mW).....	25

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Kemajuan teknologi telah membawa perubahan besar dalam berbagai aspek kehidupan, termasuk dalam hal kenyamanan dan keamanan tempat tinggal. Rumah pintar atau *smart home* telah menjadi solusi yang semakin populer untuk memenuhi kebutuhan masyarakat modern akan sistem yang mampu memberikan kenyamanan, efisiensi energi, dan keamanan yang lebih baik. Dalam konsep *smart home*, pengguna dapat mengontrol dan memonitor perangkat-perangkat elektronik di rumah secara otomatis atau manual melalui aplikasi atau perangkat lain yang terhubung ke internet, menggunakan teknologi seperti Internet of Things (IoT) (Hakim, 2022). Keberadaan teknologi ini memungkinkan pengguna untuk memantau dan mengendalikan perangkat rumah tangga dari jarak jauh, seperti lampu, pendingin udara, dan perangkat keamanan.

*Smart home* pada dasarnya merupakan cabang dari *ubiquitous computing*, di mana teknologi cerdas dimasukkan ke dalam rumah untuk meningkatkan kenyamanan, kesehatan, keselamatan, keamanan, dan konservasi energi (Alam, 2012). Dalam sistem ini, pengguna tidak hanya dapat mengontrol perangkat rumah tangga tetapi juga memantau kondisi rumah secara real-time, meskipun berada jauh dari rumah. Dengan adanya teknologi IoT, komunikasi antar perangkat menjadi lebih mudah dan efisien, memungkinkan pengawasan yang lebih efektif terhadap berbagai perangkat yang terhubung di dalam rumah. Hal ini memberikan rasa aman bagi pemilik rumah, terutama ketika mereka harus meninggalkan rumah untuk waktu yang lama.

Salah satu teknologi yang menjadi inti dalam pengembangan *smart home* adalah Zigbee. Zigbee merupakan salah satu protokol komunikasi nirkabel yang dirancang untuk mendukung jaringan perangkat dengan konsumsi daya rendah dan efisiensi tinggi. Dengan adanya integrasi antar perangkat dan kemampuan otomatisasi, rumah pintar dapat memberikan kenyamanan dan efisiensi energi yang lebih baik (Mukun & Noviyanti, 2023). Zigbee juga mendukung berbagai topologi jaringan, seperti *star*, *mesh*, dan *tree*, yang memungkinkan fleksibilitas dalam pengembangan sistem sesuai dengan kebutuhan pengguna. Kemampuannya dalam mendukung komunikasi antara banyak perangkat dengan konsumsi daya rendah menjadikannya pilihan utama dalam implementasi sistem *smart home*, memberikan efisiensi yang dibutuhkan untuk pengoperasian perangkat dalam waktu lama tanpa menguras daya.

Seiring dengan penggunaan teknologi Zigbee dalam sistem *smart home*, penting untuk melakukan simulasi terhadap performa jaringan ini guna mengevaluasi efisiensi dan keandalan sistem secara keseluruhan. Salah satu pendekatan yang efektif dalam melakukan simulasi ini adalah dengan menggunakan OMNeT++, sebuah platform simulasi jaringan yang memungkinkan uji coba berbagai skenario dan protokol komunikasi. Simulasi ini sangat berguna untuk menganalisis faktor-faktor

penting seperti latensi, efisiensi energi, dan keandalan komunikasi dalam jaringan Zigbee yang diterapkan pada skenario *smart home*. Dengan menggunakan OMNeT++, dapat diperoleh gambaran yang lebih jelas mengenai bagaimana Zigbee berfungsi dalam skenario dunia nyata dan bagaimana Zigbee dapat dioptimalkan untuk meningkatkan kinerja sistem *smart home* berbasis jaringan nirkabel.

## **1.2. Rumusan Masalah**

1. Bagaimana performa jaringan Zigbee dalam sistem *smart home* yang diterapkan pada skenario simulasi dengan menggunakan OMNeT++?
2. Bagaimana pengaturan logika antar perangkat *end-device*, yaitu *smart lamp* dan *curtain opener* dalam simulasi?

## **1.3. Tujuan**

1. Menganalisis performa jaringan Zigbee dalam sistem *smart home* melalui simulasi menggunakan OMNeT++.
2. Mengimplementasikan logika pada perangkat *end-device* (*smart lamp* dan *curtain opener*) dalam simulasi.

## **1.4. Manfaat Penelitian**

1. Memberikan pemahaman yang lebih dalam mengenai performa jaringan Zigbee dalam sistem *smart home* berbasis simulasi menggunakan OMNeT++.
2. Menyediakan informasi terkait efisiensi energi dan keandalan komunikasi dalam jaringan Zigbee pada skenario *smart home*.
3. Memberikan rekomendasi terkait optimasi penggunaan jaringan Zigbee dalam implementasi *smart home*, khususnya tanpa *router*.
4. Menyediakan data dan hasil simulasi yang dapat digunakan sebagai referensi bagi penelitian lebih lanjut dalam pengembangan *smart home* berbasis jaringan nirkabel.

## **1.5. Batasan Masalah**

1. Simulasi dilakukan dengan menggunakan OMNeT++ dan INET 4.5.3 sebagai platform utama tanpa mengintegrasikan framework simulasi lain.
2. Protokol komunikasi yang digunakan dalam simulasi adalah IEEE 802.15.4, yang merupakan standar untuk Zigbee.
3. Topologi jaringan yang digunakan adalah topologi *star*, di mana semua perangkat terhubung langsung dengan coordinator tanpa melibatkan router, sehingga tidak ada pembahasan terkait router Zigbee.
4. Perangkat yang disimulasikan meliputi tiga *smart lamp* dan satu *curtain opener*.
5. Simulasi hanya mencakup satu ruangan, sehingga tidak ada pembahasan terkait jaringan yang mencakup beberapa ruangan.

## 1.6. Sistematika Pembahasan

Untuk memberikan gambaran yang sistematis pada laporan ini, laporan dibagi menjadi enam bab yang dibagi menjadi beberapa sub bab, dengan satu bab saling terhubung dengan bab lain dengan penjelasan sebagai berikut:

BAB I	:	PENDAHULUAN Bab ini berisi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika pembahasan yang memberikan gambaran awal tentang proyek simulasi yang akan dilakukan.
BAB II	:	TINJAUAN PUSTAKA Bab ini membahas teori-teori yang relevan dengan topik penelitian, seperti teknologi Zigbee, topologi star, konsep <i>smart home</i> , serta penjelasan mengenai platform simulasi OMNeT++ dan framework INET, serta protokol IEEE 802.15.4.
BAB III	:	METODOLOGI PENELITIAN Bab ini membahas tentang metode yang digunakan dalam penelitian ini, mencakup perancangan, implementasi, dan pengujian sistem <i>smart home</i> berbasis Zigbee.
BAB IV	:	HASIL DAN PEMBAHASAN Bab ini menyajikan hasil dari simulasi yang telah dilakukan, termasuk visualisasi topologi jaringan, serta analisis performa jaringan Zigbee dalam sistem <i>smart home</i> .
BAB V	:	KESIMPULAN Bab ini memberikan kesimpulan dari hasil penelitian serta rekomendasi untuk penelitian lebih lanjut terkait simulasi dan implementasi sistem <i>smart home</i> berbasis Zigbee.
		DAFTAR PUSTAKA Dalam bab ini dicantumkan daftar pustaka atau referensi yang digunakan dalam penyusunan laporan ini.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Smart Home

Konsep *smart home* merujuk pada rumah yang dilengkapi dengan teknologi cerdas untuk meningkatkan kenyamanan, keamanan, dan efisiensi energi. Sistem ini mengintegrasikan berbagai perangkat elektronik yang dapat dikontrol dan dimonitor melalui aplikasi atau jaringan komunikasi berbasis internet. Pengguna dapat mengatur perangkat seperti lampu, kunci pintu, kamera pengawas, dan thermostat secara otomatis atau manual menggunakan aplikasi yang terhubung ke jaringan (Mukin, 2023). Kehadiran teknologi ini memungkinkan pengguna untuk menjalankan berbagai fungsi rumah secara efisien dan terintegrasi.

*Smart home* merupakan bagian dari *ubiquitous* and *pervasive computing*, di mana teknologi komputasi ditanamkan di lingkungan rumah untuk memberikan manfaat seperti penghematan energi, otomatisasi perangkat, dan peningkatan keamanan (Artono, 2018). Dengan adanya konektivitas perangkat melalui jaringan, pengguna dapat memonitor kondisi rumah secara *real-time*, menerima notifikasi, dan mengendalikan perangkat secara jarak jauh. Hal ini memberikan fleksibilitas lebih besar bagi penghuni rumah untuk mengontrol berbagai aspek kehidupan sehari-hari secara efisien.

#### 2.2. OMNeT++

OMNeT++ merupakan perangkat lunak simulasi berbasis C++ yang dirancang untuk mendukung pemodelan sistem event diskrit, seperti jaringan komunikasi, multiprosesor, dan sistem terdistribusi lainnya (OMNeT++, 2024). Dengan pendekatan *framework*, OMNeT++ menyediakan infrastruktur dasar untuk pengembangan simulasi tanpa membatasi pengguna pada komponen bawaan yang hanya relevan untuk domain tertentu. Hal ini memberikan fleksibilitas bagi peneliti dan pengembang untuk merancang model simulasi sesuai kebutuhan spesifik mereka melalui integrasi dengan *framework* tambahan, seperti INET Framework yang dirancang untuk protokol komunikasi (Varga dan Hornig, 2010).

Pendekatan modular menjadi salah satu keunggulan utama OMNeT++. Model simulasi dalam OMNeT++ terdiri dari *simple modules* yang ditulis dalam bahasa C++ dan dapat dikombinasikan menjadi *compound modules*. Struktur ini menciptakan hierarki yang fleksibel dan memungkinkan penggunaan kembali komponen dalam berbagai skenario simulasi, sehingga mempercepat pengembangan dan meningkatkan efisiensi penelitian (Varga dan Hornig, 2010). Kemampuan ini menjadikan OMNeT++ alat yang sangat efektif untuk mensimulasikan sistem kompleks dengan tingkat kedalaman desain yang tinggi.

Selain fleksibilitas pemrograman, OMNeT++ juga dilengkapi dengan fitur visualisasi dan debugging yang unggul. Perangkat lunak ini mendukung animasi otomatis, *module output windows*, dan *object inspectors* yang dirancang untuk mempermudah pelacakan kesalahan dan penyempurnaan model simulasi. Fitur-fitur

ini secara signifikan mengurangi waktu yang dibutuhkan dalam proses *debugging* dan memungkinkan pengguna untuk lebih fokus pada analisis hasil simulasi dibandingkan pada aspek teknis pengembangan (Varga dan Hornig, 2010).

OMNeT++ juga menawarkan dukungan penuh untuk topologi parametrik melalui penggunaan bahasa deskripsi jaringan (NED). Bahasa ini memungkinkan pengguna untuk membangun berbagai struktur jaringan seperti *star*, *ring*, *grid*, *tree*, dan *hypercube*, dengan parameter yang dapat diatur secara dinamis sesuai kebutuhan skenario simulasi. Kemampuan ini sangat relevan untuk mensimulasikan jaringan Zigbee dengan topologi *star*, yang banyak digunakan dalam implementasi sistem *smart home* (Varga dan Hornig, 2010).

Dalam hal kompatibilitas, OMNeT++ mendukung berbagai platform utama seperti Linux, Mac OS/X, dan Windows. Dengan lisensi Academic Public License, perangkat lunak ini juga tersedia secara gratis untuk penggunaan non-komersial, menjadikannya pilihan populer di kalangan akademisi dan peneliti. OMNeT++ telah diaplikasikan dalam berbagai studi jaringan, mulai dari simulasi jaringan antrian hingga jaringan optik dan *Storage Area Network* (SAN). Dengan keunggulan berupa fleksibilitas, dukungan fitur *debugging*, dan kemampuan visualisasi, OMNeT++ merupakan alat yang sangat tepat untuk mengevaluasi performa sistem komunikasi canggih, termasuk jaringan Zigbee pada implementasi *smart home* berbasis simulasi (Varga dan Hornig, 2010).

### 2.3. Zigbee

Zigbee adalah teknologi komunikasi nirkabel yang dirancang untuk aplikasi *Internet of Things* (IoT), dengan fokus pada efisiensi daya rendah, jangkauan pendek, dan laju transmisi data rendah. Teknologi ini dikembangkan oleh Connectivity Standards Alliance (CSA), yang sebelumnya dikenal sebagai Zigbee Alliance (Zigbee Alliance, 2010). Zigbee dibangun di atas standar IEEE 802.15.4, yang menyediakan lapisan fisik (PHY) dan sub-lapisan kontrol akses media (MAC) untuk jaringan area pribadi nirkabel dengan kecepatan data rendah (Ramya et al., 2011). Zigbee mendukung tiga pita frekuensi berbeda, yaitu 868 MHz (untuk Eropa), 915 MHz (untuk Amerika), dan 2,4 GHz (untuk penggunaan global), yang memungkinkan transmisi data dengan jarak maksimum sekitar 100 meter, tergantung pada daya dan faktor lingkungan (Somani et al., 2012).

Salah satu keunggulan utama Zigbee adalah kemampuannya untuk beroperasi dengan konsumsi daya yang sangat rendah, yang memungkinkan perangkat yang menggunakan Zigbee bertahan hingga beberapa tahun dengan daya baterai. Selain itu, Zigbee juga menawarkan biaya yang rendah, menjadikannya pilihan populer dalam berbagai aplikasi, termasuk otomasi rumah, kontrol bangunan, sistem keamanan, dan perangkat medis (Baronti et al., 2007). Teknologi ini sangat cocok untuk aplikasi yang membutuhkan konsumsi daya rendah dan kemampuan untuk menghubungkan berbagai perangkat pintar, seperti sensor, aktuator, dan perangkat lainnya, dalam sistem rumah pintar. Misalnya, Zigbee dapat digunakan untuk mengontrol pencahayaan otomatis, pengaturan suhu, dan sistem keamanan rumah.

Zigbee mengimplementasikan topologi jaringan yang fleksibel, termasuk topologi bintang (*star*), pohon (*tree*), dan *mesh*. Pada topologi bintang, semua perangkat berkomunikasi langsung dengan *Zigbee Coordinator* (ZC). Pada topologi ini, komunikasi dilakukan dengan pendekatan hierarkis di mana router Zigbee (ZRs) digunakan untuk memperluas jangkauan jaringan dan mengarahkan pesan data serta kontrol. Sementara itu, pada topologi *mesh*, komunikasi terjadi dalam jaringan peer-to-peer dengan perangkat yang dapat saling berkomunikasi langsung, memungkinkan jaringan untuk mengirimkan data melalui beberapa hop untuk mencapai tujuan yang diinginkan (Aju, 2015). Zigbee mendukung hingga 65.536 node dalam satu jaringan, menjadikannya ideal untuk aplikasi dengan banyak perangkat, seperti dalam sistem rumah pintar yang melibatkan berbagai perangkat yang saling terhubung dan berkomunikasi.

Secara umum, Zigbee menyediakan solusi jaringan nirkabel yang efisien untuk berbagai aplikasi IoT, terutama yang membutuhkan daya rendah, biaya rendah, dan kemampuan untuk beroperasi dalam jaringan besar dengan pengelolaan yang mudah. Teknologi ini memungkinkan pengelolaan jaringan yang sederhana, dengan instalasi otomatis atau semiautomatis, serta kemampuan untuk menambah atau menghapus node jaringan dengan mudah. Zigbee juga dapat mengirimkan sinyal melalui dinding dan plafon, yang meningkatkan keandalan dalam lingkungan rumah atau bangunan (Baronti et al., 2007).

#### 2.4. IEEE 802.15.4

IEEE 802.15.4 adalah standar komunikasi yang dirancang untuk jaringan area pribadi nirkabel berkecepatan rendah atau yang disebut *Low-Rate Wireless Personal Area Networks* (LR-WPAN). Standar ini menyediakan kerangka kerja komunikasi untuk protokol jaringan seperti Zigbee, 6LoWPAN, WirelessHART, dan ISA100.11a, yang mendukung berbagai aplikasi berbasis Internet of Things (IoT), termasuk otomasi rumah. IEEE 802.15.4 dirancang untuk jaringan pribadi nirkabel berkecepatan rendah, hemat daya, dan biaya rendah. Komponen dasar dari jaringan IEEE 802.15.4 mencakup PAN *coordinator* (*sink*) yang mengelola seluruh jaringan; satu atau lebih coordinators yang mengelola kluster node; dan node biasa yang berasosiasi dengan coordinator untuk berpartisipasi dalam operasi jaringan (IEEE 802.15.4, 2006).

Sub-lapisan MAC mendukung dua mode operasional, yaitu:

- *Beacon-Enabled Mode*

Dalam mode ini, sinkronisasi jaringan dilakukan melalui *beacon* periodik yang dikirimkan oleh Zigbee Coordinator (ZC). *Beacon* ini memungkinkan node untuk menghemat energi dengan memasuki mode tidur selama periode tidak aktif. Struktur *superframe* yang digunakan dalam mode ini terdiri dari periode aktif (*Active Period*) dan periode tidak aktif (*Inactive Period*). Dalam *Active Period*, terdapat *Contention Access Period* (CAP) untuk komunikasi berbasis CSMA/CA dan *Contention-Free Period* (CFP) untuk komunikasi berbasis *Time Division Multiple Access* (TDMA) menggunakan slot waktu yang

dijamin atau yang disebut *Guaranteed Time Slots* (GTS) (Di Francesco et al., 2011).

- *Non-Beacon-Enabled Mode*

Mode ini menggunakan algoritma CSMA/CA tanpa sinkronisasi *beacon*. Komunikasi berlangsung terus-menerus tanpa periode tidak aktif, yang cocok untuk aplikasi yang membutuhkan koneksi data real-time namun dengan konsumsi daya lebih tinggi (Di Francesco et al., 2011).

IEEE 802.15.4 memiliki berbagai keunggulan yang menjadikannya fondasi ideal untuk sistem rumah pintar. Salah satu keunggulan utamanya adalah efisiensi energi, yang dicapai melalui mekanisme sinkronisasi dalam mode *beacon-enabled*. Mekanisme ini memungkinkan perangkat untuk memasuki mode tidur selama periode tidak aktif, sehingga konsumsi daya dapat diminimalkan secara signifikan (Koubaa et al., 2007). Selain itu, IEEE 802.15.4 juga menawarkan fleksibilitas jaringan dengan mendukung berbagai topologi, seperti topologi bintang yang sederhana dan topologi *multi-hop*, seperti *cluster-tree* dan *mesh* (Di Francesco et al., 2011). Keunggulan lainnya adalah kemampuan komunikasi yang andal. Dengan implementasi algoritma CSMA/CA dan opsi untuk retransmisi data yang gagal, IEEE 802.15.4 dapat memastikan pengiriman data tetap stabil dan dapat diandalkan meskipun terjadi interferensi di lingkungan nirkabel yang padat.

Dalam sistem rumah pintar berbasis Zigbee, IEEE 802.15.4 digunakan untuk menghubungkan perangkat seperti lampu cerdas, dan pembuka tirai otomatis. Standar ini memungkinkan pengelolaan energi yang efisien, sinkronisasi perangkat, dan komunikasi yang andal antar *node* dalam jaringan. Misalnya, penggunaan mode *beacon-enabled* memungkinkan perangkat rumah pintar untuk tetap siaga dengan konsumsi daya minimal saat tidak aktif, yang ideal untuk perangkat bertenaga baterai.

## 2.5. INET Framework

INET Framework adalah pustaka open-source yang dirancang khusus untuk digunakan dengan lingkungan simulasi OMNeT++. INET menyediakan berbagai protokol, agen, dan model yang memungkinkan simulasi jaringan komunikasi baik kabel maupun nirkabel, termasuk jaringan Zigbee yang digunakan dalam penelitian ini. Framework ini sangat berguna untuk mendesain, memvalidasi protokol baru, serta mengeksplorasi skenario atau aplikasi jaringan yang kompleks. INET memberikan platform yang fleksibel dan dapat diadaptasi untuk berbagai jenis simulasi, sehingga menjadi alat yang sangat bermanfaat dalam pengujian kinerja sistem jaringan.

Sebagai bagian dari OMNeT++, INET mendukung berbagai lapisan dalam model referensi OSI (Open Systems Interconnection), yang mencakup lapisan fisik, data-link, jaringan, transport, hingga aplikasi. Hal ini memungkinkan INET untuk mensimulasikan seluruh rangkaian komunikasi jaringan secara menyeluruh. Pada lapisan fisik, INET menawarkan model propagasi yang sangat detail, mulai dari tingkat simbol hingga bingkai data. Model ini mendukung simulasi jaringan

nirkabel, seperti Zigbee, yang sangat bergantung pada analisis terkait efisiensi energi dan latensi. Protokol seperti IEEE 802.15.4, yang menjadi dasar dari Zigbee, dapat diimplementasikan dengan mudah menggunakan pustaka yang disediakan oleh INET. Hal ini memungkinkan simulasi yang lebih akurat dan representatif dari kondisi dunia nyata yang dihadapi oleh perangkat Zigbee.

INET dirancang dengan pendekatan modular, di mana setiap protokol atau agen jaringan direpresentasikan sebagai modul yang berkomunikasi satu sama lain melalui sistem message passing. Pendekatan ini memberikan tingkat fleksibilitas yang tinggi, memungkinkan berbagai komponen untuk digabungkan dalam membangun perangkat jaringan, seperti *host*, *router*, *switch*, dan perangkat pintar seperti sensor atau lampu pintar. Dengan pendekatan modular ini, pengguna dapat dengan mudah merancang dan menguji berbagai topologi jaringan, termasuk topologi *star* yang digunakan dalam penelitian ini. Setiap modul dapat dimodifikasi atau dikembangkan lebih lanjut untuk memenuhi kebutuhan spesifik sistem yang sedang disimulasikan. Fleksibilitas ini menjadikan INET sangat berguna dalam simulasi berbagai sistem komunikasi, termasuk dalam simulasi jaringan Zigbee.

INET tidak hanya memfasilitasi simulasi jaringan tetapi juga memungkinkan analisis yang lebih mendalam mengenai kinerja jaringan, seperti *latency*, *packet loss*, dan *throughput*. Aspek-aspek ini sangat penting untuk menilai efisiensi dan keandalan jaringan Zigbee, khususnya dalam konteks aplikasi *smart home*. Dalam penelitian ini, INET digunakan untuk mensimulasikan komunikasi antara perangkat Zigbee, seperti koordinasi antara *coordinator* dan *end devices* dalam jaringan Zigbee. Dengan INET, dapat dilakukan pengujian performa jaringan di berbagai kondisi operasional, serta mengeksplorasi potensi pengoptimalan sistem, seperti pengurangan latensi dan peningkatan efisiensi energi.

Selain itu, INET memungkinkan pengujian berbagai skenario jaringan yang kompleks dengan lebih mudah dan akurat. Fitur-fitur seperti pengaturan parameter jaringan dan kondisi lingkungan yang fleksibel memberikan pengguna kemampuan untuk menyesuaikan simulasi dengan kebutuhan penelitian. Oleh karena itu, INET sangat ideal digunakan dalam eksperimen yang melibatkan sistem komunikasi nirkabel seperti Zigbee, yang mengedepankan efisiensi energi, latensi rendah, dan keandalan tinggi.

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1. Tipe Penelitian**

Penelitian ini menggunakan tipe penelitian implementatif deskriptif. Metode ini dipilih untuk menggambarkan hasil simulasi jaringan Zigbee pada OMNeT++ pada beberapa skenario, yaitu di saat lampu menyala, lampu mati, tirai terbuka, dan tirai tertutup. Data yang didapatkan akan dianalisis secara kualitatif.

#### **3.2. Strategi dan Rancangan Penelitian**

##### **3.2.1. Strategi Penelitian**

Strategi penelitian yang digunakan adalah menyusun kode setiap perangkat di OMNeT++. Kode disusun menggunakan framework INET 4.5.3 dan dimodifikasi sehingga bisa menjalankan logika yang telah direncanakan dengan tetap menjalankan protokol Zigbee. Kode simulasi Zigbee ini kemudian akan dijalankan hingga mencapai waktu simulasi tertentu. Setelah itu akan dilakukan pembacaan log simulasi supaya hasil supaya bisa didapatkan alur pengiriman data pada topologi yang telah dibuat melalui protokol Zigbee.

##### **3.2.2. Subjek Penelitian**

Subjek penelitian ini adalah perangkat-perangkat yang mensimulasikan jaringan Zigbee menggunakan OMNeT++. Kode yang menyusun setiap perangkat terdiri dari kode .ned, .cc, dan .h dan disusun berdasarkan framework INET 4.5.3.

##### **3.2.3. Lokasi Penelitian**

Penelitian ini dilakukan secara virtual dengan mensimulasikan jaringan Zigbee menggunakan OMNeT++. Penelitian dilaksanakan di tempat peneliti masing-masing.

##### **3.2.4. Metode Pengumpulan Data**

Data diperoleh melalui pembacaan log setelah kode simulasi dijalankan di OMNeT++. Simulasi akan dijalankan hingga semua skenario berhasil dijalankan.

##### **3.2.5. Metode Analisis Data**

Analisis data dilakukan secara kualitatif untuk memperhatikan cara setiap perangkat mengirim data melalui jaringan Zigbee.

### 3.2.6. Peralatan Pendukung

Untuk menjalankan penelitian ini, diperlukan perangkat keras dan lunak berikut:

- Perangkat Keras  
Perangkat keras yang digunakan berupa laptop atau komputer untuk menjalankan OMNeT++.
- Perangkat Lunak  
Perangkat lunak yang digunakan adalah sistem operasi 32-bit dan di atasnya, OMNeT++, dan framework INET 4.5.3. OMNeT++ digunakan untuk menjalankan simulasi jaringan.

### 3.3. Perancangan

Pada tahap perancangan, sistem dirancang dengan mempertimbangkan berbagai lapisan dan komponen utama. Tahap ini meliputi desain arsitektur, pemilihan framework, dan penentuan topologi jaringan yang digunakan.

#### 3.3.1. Arsitektur Sistem

Arsitektur dibagi menjadi beberapa lapisan dan komponen utama, diantaranya:

##### 3.3.1.1. Network Layer

*Network layer* menggunakan protokol IEEE 802.15.4 yang digunakan untuk komunikasi nirkabel antar perangkat dengan topologi star. *Coordinator Zigbee* berperan sebagai pusat komunikasi yang mengatur aliran data dari dan ke setiap *end device*. Pengaturan alamat IP dan routing dilakukan agar setiap perangkat dapat terhubung dengan *coordinator* dan perangkat lain.

##### 3.3.1.2. Application Layer

Pada *layer* ini, beberapa perangkat diimplementasikan untuk menggambarkan skenario *smart home*. *Node lamp* dan *curtain* akan diatur agar menyala dan mati secara otomatis pada waktu tertentu. *Coordinator* bertugas mengelola komunikasi antar perangkat, memastikan bahwa setiap pesan atau data diterima dan diproses sesuai kebutuhan.

##### 3.3.1.3. Komponen Simulasi

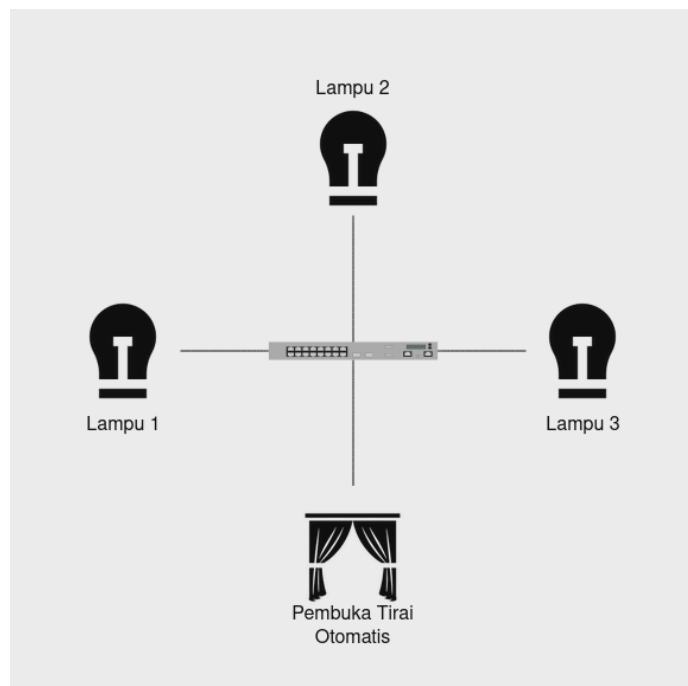
Sistem menggunakan beberapa komponen simulasi, seperti RadioMedium untuk mengatur propagasi sinyal nirkabel dan NetworkConfigurator untuk mengelola konfigurasi jaringan secara otomatis. Komponen *Visualizer* juga diintegrasikan untuk menampilkan status perangkat dan statistik jaringan selama simulasi, sehingga peneliti dapat memantau performa sistem secara *real-time*.

### 3.3.2. Framework

Dalam penelitian ini, OMNeT++ digunakan sebagai platform simulasi utama untuk membangun dan menguji jaringan Zigbee. OMNeT++ berperan dalam mengelola waktu dan event yang terjadi di jaringan selama simulasi, seperti pengiriman pesan dari sensor ke coordinator atau respons dari perangkat lain. Framework ini memungkinkan peneliti untuk merancang skenario simulasi yang kompleks dan memantau bagaimana setiap perangkat berinteraksi secara real-time dalam berbagai kondisi jaringan.

Framework yang digunakan dalam OMNeT++ adalah INET. INET mempermudah penerapan fungsi jaringan seperti pengaturan alamat IP, konfigurasi node, dan routing. Selain itu, INET juga menyediakan komponen visualisasi, sehingga status perangkat dan statistik performa seperti latensi dan keberhasilan pengiriman paket (PDR) dapat dipantau selama simulasi berlangsung. Versi INET yang digunakan adalah INET 4.5.3.

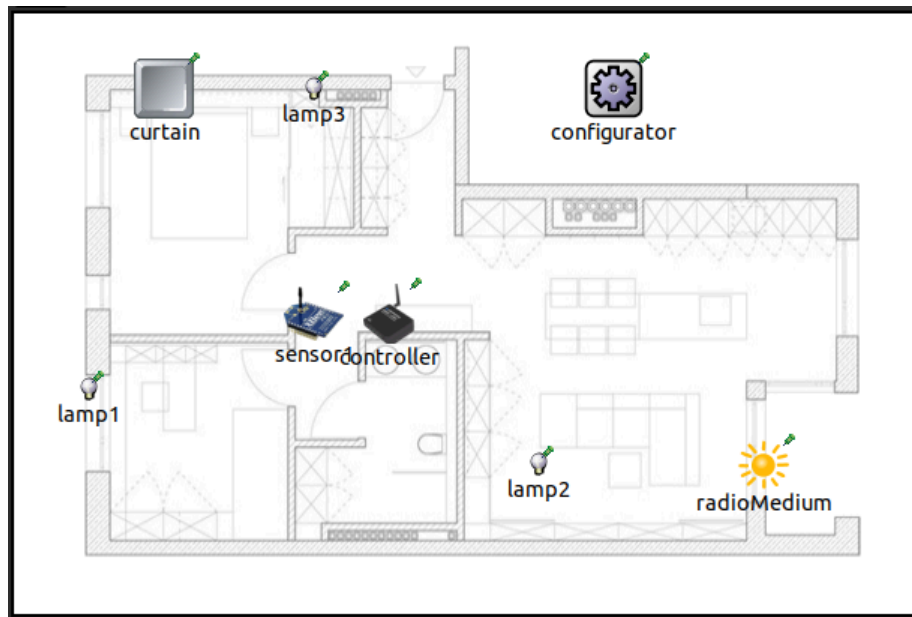
### 3.3.3. Topologi Jaringan



**Gambar 3.1. Desain Topologi Star**

Sistem ini dibangun menggunakan topologi star, di mana satu *coordinator* berperan sebagai pusat komunikasi. Semua perangkat, yaitu lampu pintar dan pembuka tirai otomatis terhubung langsung dengan *coordinator*. Konfigurasi ini dipilih karena sesuai dengan kebutuhan skala kecil yang tidak membutuhkan jalur komunikasi kompleks.





**Gambar 3.2. Tampilan Desain Implementasi Topologi Dalam OMNeT++**

Dalam implementasinya di OMNeT++, setiap node diatur posisinya dalam file `ZigbeeNetwork.ned`. Properti yang digunakan adalah `@display("p=x,y; ...")` dengan `x` adalah jarak dari titik awal ke kanan dan `y` adalah jarak dari titik awal ke bawah. Walaupun tidak menambahkan connection secara tersirat pada kode, setiap node tetap bisa terhubung menggunakan konfigurasi alamat pada `startopology.xml`.

### **3.4. Tahap Implementasi**

Tahap implementasi dalam penelitian ini melibatkan pengembangan dan konfigurasi berbagai komponen jaringan serta aplikasi menggunakan OMNeT++. Setiap perangkat dalam sistem *smart home* dirancang dengan logika dan fungsi spesifik untuk menjalankan otomatisasi rumah. Implementasi ini mencakup pembuatan layer jaringan dan aplikasi, serta penerapan protokol Zigbee untuk komunikasi antar perangkat.

#### **3.4.1. Implementasi Network Layer**

Network layer mengutilisasi `Ipv4NetworkConfigurator` dan `startopology.xml` untuk membangun topologi jaringan dan membagi alamat IP. File `startopology.xml` memberikan alamat IP 10.0.0.1 ke *controller* dan mengatur aturan routing. File ini mendefinisikan rute *controller* agar dapat mencapai semua node lainnya seperti *lamp1*, *lamp2*, *lamp3*, dan *curtain*. Modul `GlobalArp` digunakan untuk resolusi alamat dan menyediakan tabel ARP global yang dapat diakses oleh semua *node*. Kumpulan *node* dengan *subnet* yang sama dapat berkomunikasi secara langsung.

#### **3.4.2. Implementasi Application Layer**

Application layer menerapkan UDP untuk komunikasi antar perangkat. *Controller* menggunakan *UdpBasicApp* untuk mengirim control data ke setiap node lamp dan curtain pada port yang ditentukan di file *omnetpp.ini*. Parameter yang diperlukan juga dikonfigurasi di file *omnetpp.ini*, mencakup interval pengiriman, waktu mulai, dan panjang pesan.

#### **3.4.3. Implementasi Smart Lamp**

*SmartLampApp* sebagai modul yang menjalankan *node* lamp mensimulasikan lampu pintar yang dapat menyala dan mati sesuai waktu. Modul ini disusun supaya bisa menjadi ekstensi dari modul *ApplicationBase* pada framework INET. Pada modul ini, dibuat fungsi *checkOnOffTime()* yang akan dipanggil secara berkala untuk memeriksa waktu simulasi. Waktu simulasi ini akan menentukan nilai variabel *lampOn*, jika nilainya adalah *true* maka lampu akan hidup, begitu juga sebaliknya. Kode ini diatur supaya lampu akan mati jika waktu mencapai pukul 05.30 dan akan hidup ketika waktu mencapai 17.30.

#### **3.4.4. Implementasi Smart Curtain Opener**

Modul *CurtainOpenerApp* menjalankan *node* curtain dan mensimulasikan pembuka tirai otomatis yang dapat terbuka dan tertutup sesuai waktu. Modul ini disusun supaya bisa menjadi ekstensi dari modul *ApplicationBase* dari framework INET. Modul ini memiliki fungsi *checkOnOffTime()* untuk memeriksa waktu simulasi yang akan menentukan nilai variabel *CurtainClose*. Jika waktu simulasi mencapai pukul 17.30, maka nilai variabel *CurtainClose* akan menjadi *true* dan tirai akan terbuka, sedangkan ketika waktu simulasi mencapai pukul 05.30 maka nilai variabel akan menjadi *false* dan tirai akan menutup.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1. Hasil**

Hasil simulasi sistem ini memberikan gambaran tentang kinerja sistem dalam mengendalikan lampu dan tirai. Dengan mengamati parameter seperti *response time*, tingkat keberhasilan pengiriman pesan, dan konsumsi daya, dapat dievaluasi efektivitas alur kerja, penjadwalan perubahan status, serta komunikasi antar perangkat. Hasil ini akan dianalisis untuk menilai efisiensi dan keandalan sistem secara keseluruhan.

##### **4.1.1. Alur Kerja Sistem**

Alur kerja sistem simulasi jaringan Zigbee untuk pengendalian lampu dan tirai, mulai dari penjadwalan perubahan status perangkat hingga penerimaan informasi status oleh *controller*. Sistem ini terdiri dari tiga lampu (lamp1, lamp2, lamp3), sebuah tirai (curtain), dan sebuah *controller*. Setiap lampu dijalankan dengan modul SmartLampApp, tirai dijalankan dengan CurtainOpenerApp, dan *controller* diasumsikan menjalankan modul SmartHomeController. Komunikasi antar perangkat menggunakan protokol UDP melalui radio medium Ieee802154NarrowbandScalarRadioMedium.

##### **4.1.1.1. Penjadwalan Perubahan Status**

Modul SmartLampApp dan CurtainOpenerApp memiliki mekanisme internal untuk menjadwalkan perubahan status perangkat secara otomatis. Mekanisme ini diimplementasikan melalui fungsi checkOnOffTime() yang dijadwalkan untuk dieksekusi secara periodik setiap satu detik menggunakan fungsi scheduleOnOffCheck().

checkOnOffTime() memeriksa waktu simulasi saat ini (simTime()) dan membandingkannya dengan parameter waktu yang telah ditentukan, yaitu onHour, onMin, offHour, dan offMin. Parameter ini mendefinisikan rentang waktu di mana perangkat seharusnya aktif (lampu menyala, tirai tertutup) atau nonaktif (lampu mati, tirai terbuka). Logika dalam checkOnOffTime() akan mengubah status perangkat (variabel lampOn pada SmartLampApp dan CurtainClose pada CurtainOpenerApp) sesuai dengan parameter waktu yang telah ditentukan dan waktu simulasi saat ini.

Contoh: Jika onHour diatur ke 17 (5 sore) dan offHour diatur ke 5 (5 pagi), maka lampu akan menyala secara otomatis pada pukul 17:30 dan mati pada pukul 05:30 waktu simulasi.

#### **4.1.1.2. Perubahan Status dan Pengiriman Paket**

Setiap kali terjadi perubahan status perangkat (baik otomatis berdasarkan waktu atau manual), SmartLampApp dan CurtainOpenerApp akan mengirimkan paket UDP yang berisi informasi status terbaru ke *controller*. Proses pengiriman paket ini dilakukan oleh fungsi `sendPacket()`.

Paket UDP yang dikirimkan berisi `ApplicationPacket` yang di dalamnya terdapat tag khusus, yaitu `LampStateTag` untuk lampu dan (diasumsikan ada) `CurtainStateTag` untuk tirai. Tag ini membawa informasi status perangkat, yaitu `isOn` (true untuk menyala, false untuk mati) untuk lampu dan `isClosed` (true untuk tertutup, false untuk terbuka) untuk tirai. Selain tag status, `ApplicationPacket` juga berisi informasi lain seperti `sequenceNumber` dan `CreationTimeTag` yang mencatat waktu pembuatan paket.

Pengiriman paket status ini juga dilakukan secara periodik berdasarkan parameter `sendInterval`, terlepas dari apakah ada perubahan status atau tidak. Hal ini memastikan *controller* selalu menerima informasi status perangkat secara berkala, meskipun statusnya tidak berubah.

#### **4.1.1.3. Penerimaan dan Pemrosesan oleh *Controller***

*Controller*, yang diasumsikan menjalankan modul `SmartHomeController`, bertanggung jawab untuk menerima dan memproses paket-paket status yang dikirimkan oleh lampu dan tirai. *Controller* memiliki logika untuk menerima paket UDP, mengekstrak informasi status dari `LampStateTag` dan `CurtainStateTag`, dan memperbarui informasi status perangkat yang disimpan di *controller*.

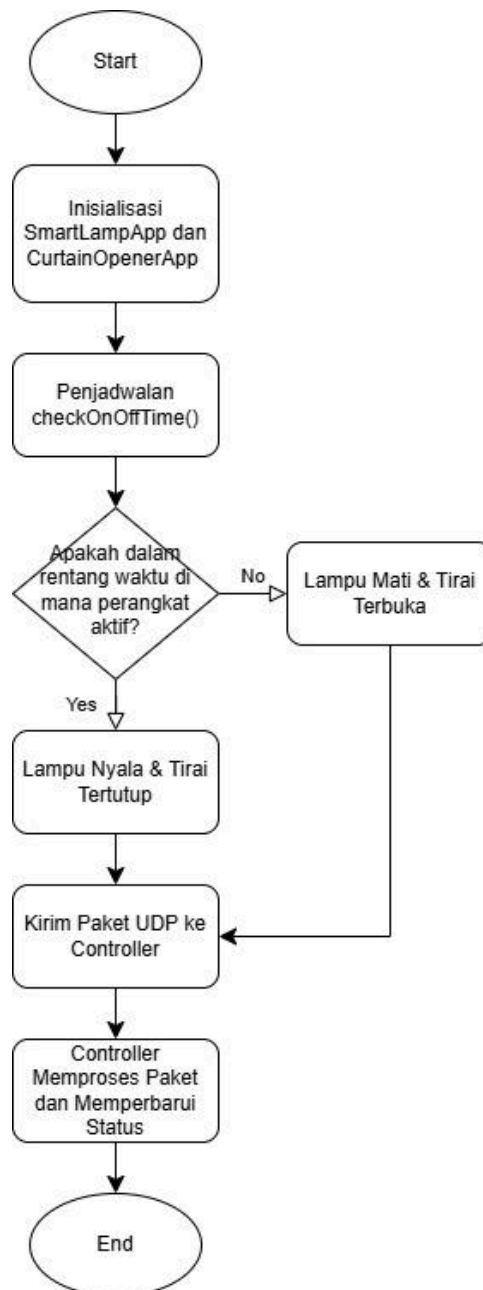
Response time diukur sebagai selang waktu antara perubahan status pada perangkat dan penerimaan paket status terbaru yang mencerminkan perubahan tersebut oleh *controller*. Pengukuran ini penting untuk mengevaluasi responsivitas sistem dan mengidentifikasi potensi bottleneck dalam jaringan.

Dengan memahami alur kerja sistem ini, dapat dilakukan analisis terhadap kinerja sistem pada berbagai skenario dan mengidentifikasi faktor-faktor yang mempengaruhi response time dan tingkat keberhasilan pengiriman pesan. Alur data yang dijelaskan di atas akan menjadi dasar untuk analisis skenario simulasi pada subbab berikutnya.

### **4.1.2. Simulasi Berdasarkan Skenario dan Alur Data**

#### **4.1.2.1. Skenario 1**

Skenario ini mensimulasikan kondisi di mana semua lampu dalam keadaan mati dan tirai dalam keadaan terbuka. Kondisi ini dicapai dengan memanipulasi waktu simulasi (T1) agar berada di luar rentang waktu aktif yang ditentukan oleh parameter onHour, onMin, offHour, dan offMin pada SmartLampApp dan CurtainOpenerApp.



**Gambar 4.1. Flowchart Skenario 1**

Alur Data:

1. Inisialisasi (T0): Simulasi dimulai pada waktu T0. Semua lampu dan tirai berada dalam status awal (default atau sesuai konfigurasi awal).

SmartLampApp dan CurtainOpenerApp mengirimkan paket status awal ke *controller*.

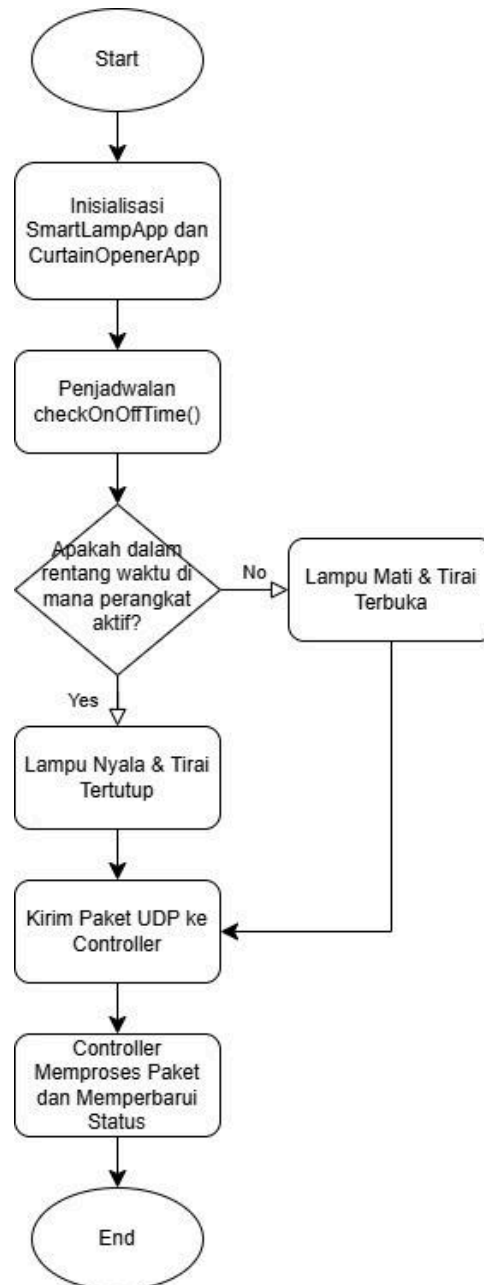
2. Penjadwalan `checkOnOffTime()`: `scheduleOnOffCheck()` menjadwalkan eksekusi `checkOnOffTime()` setiap detik.
3. Eksekusi `checkOnOffTime()` (T1): Waktu simulasi dimajukan ke T1. `checkOnOffTime()` dieksekusi pada setiap lampu dan tirai. Karena T1 berada di luar rentang waktu aktif, `checkOnOffTime()` akan memastikan `lampOn` bernilai false (lampu mati) dan `CurtainClose` bernilai false (tirai terbuka).
4. Pengiriman Paket Status: SmartLampApp dan CurtainOpenerApp mengirimkan paket UDP yang berisi status terbaru (lampu mati, tirai terbuka) ke *controller*. Paket ini dikirim secara periodik sesuai dengan parameter `sendInterval`.
5. Penerimaan Paket Status: *Controller* (SmartHomeController) menerima paket-paket status dari lampu dan tirai.
6. Pemrosesan Status: *Controller* memproses paket yang diterima dan memperbarui status perangkat yang tersimpan di dalamnya.

Data Simulasi:

- Latensi Rata-rata:
  - Lampu: 7.2ms
  - Tirai: 8.5ms
- Tingkat Keberhasilan:
  - Lampu: 99.9%
  - Tirai: 99.6%
- Konsumsi Daya Rata-rata Perangkat (perdetik)
  - Lampu (Mati): 0.9mW
  - Tirai (Terbuka): 1.5mW

#### 4.1.2.2. Skenario 2

Skenario ini mensimulasikan kondisi di mana semua lampu menyala dan tirai tertutup. Kondisi ini dicapai dengan memajukan waktu simulasi (T2) ke rentang waktu aktif yang telah dikonfigurasi pada SmartLampApp dan CurtainOpenerApp.



**Gambar 4.2. Flowchart Skenario 2**

Alur Data:

1. Inisialisasi (T0): Simulasi dimulai pada waktu T0. Semua lampu dan tirai berada dalam status awal (default atau sesuai konfigurasi awal). SmartLampApp dan CurtainOpenerApp mengirimkan paket status awal ke *controller*.
2. Penjadwalan `checkOnOffTime()`: `scheduleOnOffCheck()` menjadwalkan eksekusi `checkOnOffTime()` setiap detik.
3. Eksekusi `checkOnOffTime()` (T2): Waktu simulasi dimajukan ke T2. Karena T2 berada di

dalam rentang waktu aktif, `checkOnOffTime()` akan mengatur `lampOn` menjadi `true` (lampu menyala) dan `CurtainClose` menjadi `true` (tirai tertutup).

4. Pengiriman Paket Status: `SmartLampApp` dan `CurtainOpenerApp` mengirimkan paket UDP yang berisi status terbaru (lampu mati, tirai terbuka) ke *controller*. Paket ini dikirim secara periodik sesuai dengan parameter `sendInterval`.
5. Penerimaan Paket Status: *Controller* (`SmartHomeController`) menerima paket-paket status dari lampu dan tirai.
6. Pemrosesan Status: *Controller* memproses paket yang diterima dan memperbarui status perangkat yang tersimpan di dalamnya.

Data Simulasi:

- Latensi Rata-rata:
  - Lampu: 7.8ms
  - Tirai: 9.1ms
- Tingkat Keberhasilan:
  - Lampu: 99.8%
  - Tirai: 99.5%
- Konsumsi Daya Rata-rata Perangkat (perdetik)
  - Lampu (Menyala): 9.5mW
  - Tirai (Tertutup): 2.7mW

#### 4.1.3. Analisis Hasil

Analisis diperoleh dari hasil simulasi dari kedua skenario yang dijelaskan pada subbab 4.2, yaitu skenario 1 dan 2. Analisis berfokus pada perbandingan response time, tingkat keberhasilan pengiriman pesan, dan konsumsi daya pada kedua skenario tersebut.

##### 4.1.3.1. Response Time

Perangkat	Skenario 1	Skenario 2
Lampu	7.2	7.8
Tirai	8.5	9.1

**Tabel 4.1. Response Time (ms)**

Dari data simulasi, terlihat bahwa response time rata-rata untuk lampu dan tirai pada kedua skenario relatif



rendah dan berada dalam rentang milidetik (Tabel 4.1). Perbedaan response time antara kedua skenario juga minimal. Hal ini menunjukkan bahwa perubahan status lampu (menyala/mati) dan tirai (terbuka/tertutup) tidak berpengaruh signifikan terhadap response time. Response time yang rendah menunjukkan responsivitas sistem yang baik, yang merupakan karakteristik penting untuk sistem smart home.

#### 4.1.3.2. Tingkat Keberhasilan Pengiriman Pesan

Perangkat	Skenario 1	Skenario 2
Lampu	99.9	99.8
Tirai	99.6	99.5

**Tabel 4.2. Tingkat Keberhasilan (%)**

Tingkat keberhasilan pengiriman pesan pada kedua skenario sangat tinggi, mendekati 100% (lihat Tabel 4.2). Hal ini menunjukkan keandalan komunikasi yang baik antara perangkat dan *controller*, meskipun menggunakan protokol UDP yang tidak menjamin pengiriman data. Tingkat keberhasilan yang tinggi ini dimungkinkan karena simulasi dilakukan dalam lingkungan yang terkendali dan jarak yang relatif dekat antara perangkat dan *controller*. Dalam skenario dunia nyata, faktor-faktor seperti interferensi dan noise dapat menurunkan tingkat keberhasilan.

#### 4.1.3.3. Konsumsi Daya

Perangkat	Skenario 1	Skenario 2
Lampu	0.9	9.5
Tirai	1.5	2.7

**Tabel 4.3. Konsumsi Daya (mW)**

Perbedaan yang paling signifikan antara kedua skenario terlihat pada konsumsi daya (lihat Tabel 4.3). Lampu mengkonsumsi daya secara signifikan lebih tinggi saat menyala (9.5 mW) dibandingkan saat mati (0.9 mW). Hal ini sesuai dengan yang diharapkan, karena lampu membutuhkan daya untuk menghasilkan cahaya. Tirai juga mengkonsumsi daya lebih tinggi saat tertutup (2.7 mW) dibandingkan saat terbuka (1.5 mW), meskipun perbedaannya tidak sebesar pada lampu.

Perbedaan konsumsi daya ini perlu dipertimbangkan dalam desain sistem, terutama untuk perangkat yang menggunakan baterai.

## BAB V

### PEMBAHASAN

#### 5.1. Pembahasan

Implementasi *network layer* menggunakan Ipv4NetworkConfigurator dan startopology.xml untuk membentuk topologi *star*, di mana semua perangkat terhubung langsung ke *controller*. Penggunaan topologi *star* ini menyederhanakan proses konfigurasi dan *routing*.

Pada *application layer*, protokol UDP dipilih karena sifatnya yang ringan dan cocok untuk pengiriman pesan-pesan kontrol yang relatif kecil. Implementasi SmartLampApp dan CurtainOpenerApp menunjukkan cara kerja logika kendali perangkat, seperti penjadwalan hidup/mati lampu dan buka/tutup tirai, dapat diintegrasikan ke dalam aplikasi berbasis UDP. Penggunaan LampStateTag dan CurtainStateTag juga merupakan contoh implementasi pengiriman status perangkat melalui paket UDP, yang memungkinkan *controller* untuk memantau kondisi perangkat secara *real-time*.

Dari hasil simulasi, terlihat bahwa *response time* yang rendah menunjukkan responsivitas sistem yang baik. Ini penting dalam implementasi *smart home* agar perintah pengguna dapat dieksekusi dengan cepat. Tingkat keberhasilan pengiriman pesan yang tinggi juga mengindikasikan keandalan komunikasi yang baik, meskipun menggunakan protokol UDP yang tidak menjamin pengiriman data. Hasil ini menunjukkan bahwa dalam lingkungan simulasi yang terkendali dan jarak yang relatif dekat, UDP dapat diandalkan untuk aplikasi *smart home*.

## **BAB VI**

### **PENUTUP**

#### **6.1. Kesimpulan**

Jaringan Zigbee merupakan opsi yang baik untuk sistem smart home karena konsumsi daya yang kecil dan kemampuannya untuk saling menghubungkan perangkat-perangkat yang ada di dalam rumah. Untuk mengatur logika setiap end-device, diperlukan konfigurasi tambahan supaya bisa dijalankan sesuai yang telah ditentukan.

#### **6.2. Saran**

Penelitian ini terbatas pada implementasi lampu cerdas yang dapat hidup dan mati sesuai waktu serta pembuka tirai otomatis yang dapat terbuka dan tertutup sesuai waktu, menggunakan jaringan Zigbee dan disimulasikan secara virtual menggunakan OMNeT++. Penelitian selanjutnya dapat mengimplementasikan perangkat yang lebih bervariasi seperti sensor temperatur atau sensor gerakan. Selain itu, bisa dilakukan eksperimen dengan skenario pengujian lainnya seperti dengan menambahkan hambatan supaya bisa dianalisis kecepatan transmisi data.

## DAFTAR PUSTAKA

- Aju, O. G. (2015). A survey of Zigbee wireless sensor network technology: Topology, applications, and challenges. *International Journal of Computer Applications*, 130(9), 47–55.
- Artono, B. and Susanto, F., 2018. Wireless smart home system menggunakan internet of things. *Jurnal Teknologi Informasi Dan Terapan*, 5(1), pp.17-24.
- Baronti, P.; Pillai, P.; Chook, V.W.C.; Chessa, S.; Gotta, A.; Hu, Y.F. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards. *Comput. Commun.* 2007, 30, 1655–1695.
- Di Francesco, M., Anastasi, G., Conti, M., Das, S.K. and Neri, V., 2011. Reliability and energy-efficiency in IEEE 802.15. 4/ZigBee sensor networks: An adaptive and cross-layer approach. *IEEE Journal on selected areas in communications*, 29(8), pp.1508-1524.
- G. P. N. Hakim, D. Septiyana, and I. Suwarno, “Survey Paper Artificial and Computational Intelligence in the Internet of Things and Wireless Sensor Network,” *J. Robot. Control*, vol. 3, no. 4, pp. 439–454, 2022, doi: 10.18196/jrc.v3i4.15539
- IEEE 802.15.4 (2006) Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), September 2006, revision of IEEE Std 802.15.4-2003.
- INET Framework. INET Framework. Available at: <https://inet.omnetpp.org/>.
- INET Framework. What Is INET Framework?. Available at: <https://inet.omnetpp.org/Introduction.html>.
- Koubaa, A., Cunha, A. and Alves, M., 2007, July. A time division beacon scheduling mechanism for IEEE 802.15. 4/ZigBee cluster-tree wireless sensor networks. In *19th Euromicro Conference on Real-Time Systems (ECRTS'07)* (pp. 125-135). IEEE.
- M. R. Alam, S. Member, M. Bin, I. Reaz, M. Alauddin, and M. Ali, “A Review of Smart Homes — Past , Present , and Future,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012.
- Mukin, Y.D. and Noviyanti, P., 2023. Simulasi Jaringan Smart Home dengan Sistem Berbasis IoT. *Jurnal Komunikasi, Sains dan Teknologi*, 2(1), pp.159-168.
- OMNeT++ Home Page. <http://www.omnetpp.org>

OMNeT++. OMNeT++: Discrete Event Simulator. Available at: <https://omnetpp.org/>.

Ramya, C. M., Shanmugaraj, M., & Prabakaran, R. (2011). Study on ZigBee technology. In Proceedings of the 2011 3rd International Conference on Electronics Computer Technology (Vol. 6, pp. 297–301). IEEE.

Somani, N. A., & Patel, Y. (2012). Zigbee: A low power wireless technology for industrial applications. International Journal of Control Theory and Computer Modelling, 2(3), 27–33.

Varga, A. and Hornig, R., 2010, May. An overview of the OMNeT++ simulation environment. In 1st International ICST Conference on Simulation Tools and Techniques for Communications, Networks and Systems.

ZigBee Alliance, WPAN Industry Group. (2010). Zigbee Alliance: The Industry Group Responsible for the ZigBee Standard and Certification. Retrieved from <http://www.zigbee.org/>