

## BAB III METODE PENELITIAN

Pada penelitian ini penulis akan membuat perancangan jaringan yang digunakan untuk sistem presensi mahasiswa dengan tujuan mewujudkan *smart campus*. Perancangan jaringan tersebut berbasis alat komunikasi *ZigBee* (802.15.4). Penelitian ini dirancang dengan topologi *tree* dan akan diukur kinerja dari jaringan tersebut disuatu tempat yang terdapat interferensi dari perangkat lain. Dan hasil penelitian ini akan dibandingkan dengan topologi lainnya. Diperlukan metodologi penelitian yang digunakan pada tugas akhir ini adalah sebagai berikut.

### 3.1 Studi Literatur

Studi literatur yang digunakan pada penelitian ini mencakup teori – teori dasar yang mendukung dalam perancangan jaringan *ZigBee*, Arduino Uno, komunikasi topologi *tree*, dan parameter QOS.

### 3.2 ALUR PENELITIAN

#### 3.2.1 Flowchart Perancangan



**Gambar 3.1 Flowchart Perancangan Jaringan**

### 3.2.2 Block diagram pada jaringan komunikasi *ZigBee*

Pada komunikasi jaringan *ZigBee* topologi *tree* menggunakan beberapa *node* yang memiliki fungsi yang berbeda – berbeda. Perangkat yang digunakan pada *node end device* adalah Arduino uno dan modul *XBee* yang nantinya akan mengirimkan data yang dikirimkan. Kemudian pada router, perangkat yang digunakan adalah Arduino uno dan sebuah modul *Xbee* yang akan meneruskan data dari sebuah *end device* ke *coordinator*. Dan terakhir pada *coordinator* perangkat yang digunakan adalah Arduino uno, modul *Xbee*, dan sebuah laptop untuk melihat data yang dikirimkan dari *end device*. Berikut block diagram jaringan *ZigBee*.



**Gambar 3.2** Blok diagram Jaringan Komunikasi *ZigBee*

### 3.2.3 Penentuan Lokasi Jaringan Komunikasi *ZigBee*

Pada penelitian ini tugas akhir ini penulis menggunakan dua percobaan yang bertujuan untuk melihat pengaruh interferensi terhadap jaringan topologi *tree* yang akan digunakan. Percobaan pertama dilakukan pada kampus IT Telkom Purwokerto yang terdapat interferensi. Dan percobaan kedua akan dilakukan diruang bebas interferensi.

## 3.3 Perancangan Alat

Dalam penelitian ini, penulis membuat perancangan sebuah jaringan dengan topologi *tree* dari sistem komunikasi *ZigBee* digunakan beberapa peralatan yang digunakan yaitu:

### 3.3.1 Modul *XBee*

*Xbee* merupakan modul transmitter dan receiver yang didesain untuk memenuhi standar *ZigBee* (802.15.4). Modul ini berfungsi sebagai pengirim dan penerima. Modul *Xbee* ada 2 jenis, yaitu *Xbee series 1* dan *Xbee series 2*, antara *Xbee series 1*

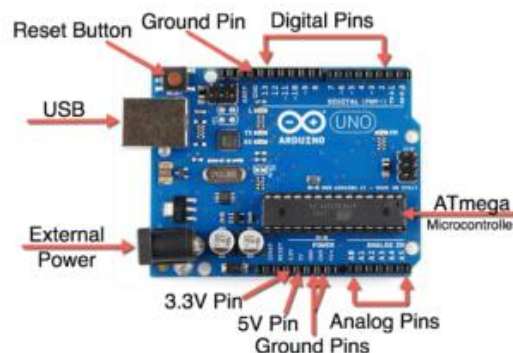
dan series 2 bisa saling terhubung. *Xbee series 1* beroperasi dengan menggunakan protokol 802.15.4(*Low-Rate Wireless Personal Networks*) sedangkan *Xbee series 2* beroperasi pada protokol *ZigBee* (protokol yang dibangun diatas 802.15.4). Pada penelitian ini penulis menggunakan *Xbee series 1*.



**Gambar 3.3 Modul XBee**

### 3.3.2 Arduino Uno

Arduino Uno adalah board mikrokontroler berbasis ATmega328 (*datasheet*). Memiliki 14 pin *input* dari *output* digital dimana 6 pin input tersebut dapat digunakan sebagai *output* PWM dan 6 pin input analog, 16 MHz osilator kristal, koneksi USB, jack power, ICSP header, dan tombol reset. Untuk mendukung mikrokontroler agar dapat digunakan, cukup hanya menghubungkan Board Arduino Uno ke komputer dengan menggunakan kabel USB atau listrik dengan AC yang-ke adaptor-DC atau baterai untuk menjalankannya.



**Gambar 3.4 Arduino Uno**

### 3.3.3 Komputer

Komputer merupakan perangkat elektronik yang dapat memanipulasi dan memproses informasi atau data. Komputer juga mampu menyimpan, mengambil, dan mengolah sebuah informasi. Pada penelitian ini komputer digunakan sebagai alat untuk mengkonfigurasi *Xbee* dan Arduino Uno, dan melihat hasil output yang dihasilkan pada jaringan *ZigBee* tersebut.

## 3.4 Perangkat Lunak untuk Penelitian

### 3.4.1 DIGI XCTU

XCTU merupakan sebuah *software* yang digunakan dalam melakukan konfigurasi *Xbee*. Software ini juga dapat digunakan untuk menentukan PAN ID, mode API, dan destination address. Tipe node dari modul *Xbee* seperti *coordinator*, *router*, dan *end device* dapat diatur juga dalam aplikasi ini.

**Tabel 3.1 Parameter yang dikonfigurasi pada XCTU**

PAN ID	PAN ID atau <i>Personal Area Network Identity Number</i> , merupakan identitas suatu jaringan. Dimana PAN ID setiap <i>node</i> harus sama agar dapat berkomunikasi satu sama lain.
<i>Destination High</i>	Merupakan 32bit alamat tujuan paket dikirim
<i>Destination Low</i>	Merupakan 32bit alamat tujuan paket dikirim. Jika di gabungkan dengan DH maka akan menghasilkan 64bit <i>address</i>
<i>Serial Number High</i>	Alamat unik 32bit yang merupakan identitas modul <i>XBee</i>
<i>Serial Number Low</i>	Alamat unik 32bit yang merupakan identitas modul <i>XBee</i> . Jika di gabungkan dengan DH maka akan menghasilkan 64bit <i>address</i>
<i>MY (Source Address)</i>	Alamat unik 16bit yang dapat dikonfigurasi secara manual.
<i>Baud Rate</i>	Jumlah bit (digit biner) yang biasa dikirim dalam satuan detik

### 3.4.2 Arduino IDE

IDE itu merupakan kependekan dari *Integrated Development Environment*, atau secara bahasa mudahnya merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Arduino IDE merupakan software yang bersifat open source untuk mengkonfigurasi papan arduino tersebut. *Software* ini dapat bekerja pada sistem operasi Windows, Macintosh, dan Linux.

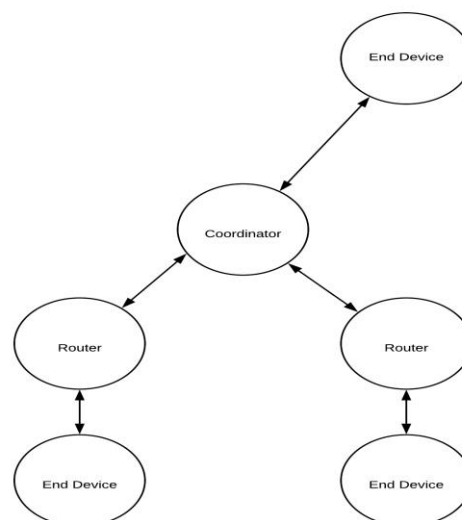
### 3.4.3 Serial Port Monitor

*Serial port monitor* merupakan sebuah software yang memiliki fungsi untuk memonitoring atau merekam data yang masuk dan keluar dari port serial. Fitur yang ada pada serial port monitor antara lain perekaman data dalam bentuk hexadecimal, adanya *timestamp* dari data yang masuk.

## 3.5 PERANCANGAN TOPOLOGI ZIGBEE

### 3.5.1 Topologi Jaringan *Tree*

Topologi jaringan *tree* menggunakan tiga buah node, yaitu *end device*, *router*, dan *coordinator*. Sehingga perangkat yang digunakan berjumlah 6 set. Dimana 3 set akan dijadikan *end device*, 2 set digunakan sebagai *router*, dan 1 set akan digunakan



sebagai *coordinator*.

### Gambar 3.5 Topologi Jaringan *Tree*

a. Konfigurasi modul *XBee* topologi *tree*

konfigurasi modul *XBee* pada *node end device* pada tabel 3.2

**Tabel 3.2 Konfigurasi *node end device***

<b>PAN ID</b>	1105
<b><i>Destination Address High</i></b>	0
<b><i>Destination Address Low</i></b>	<i>ID Coordinator / Router</i>
<b><i>Baud Rate</i></b>	9600

konfigurasi modul *XBee* pada *node router* pada tabel 3.3

**Tabel 3.3 konfigurasi *node router***

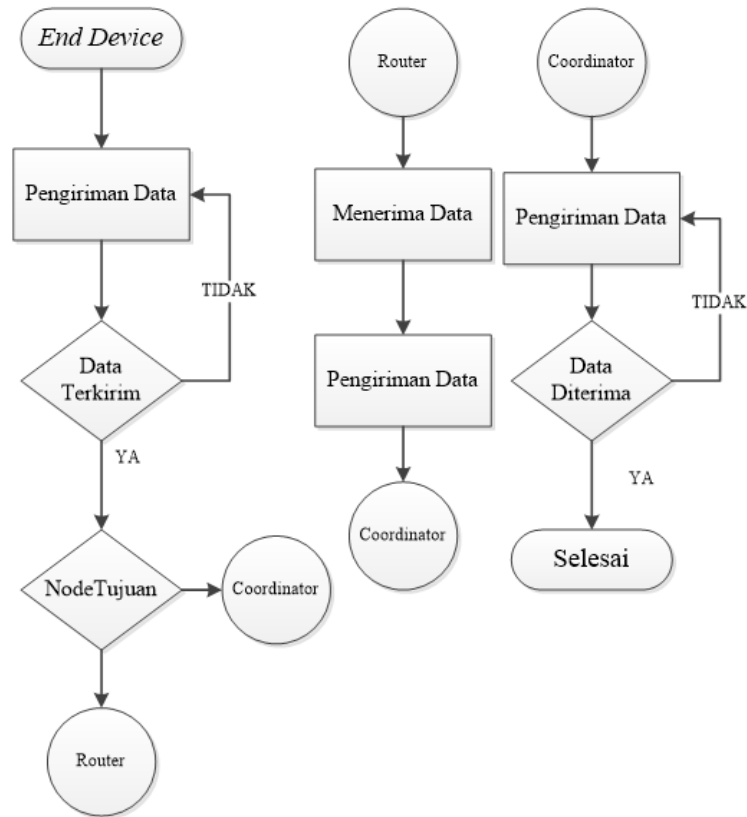
<b>PAN ID</b>	1105
<b><i>Destination Address High</i></b>	0
<b><i>Destination Address Low</i></b>	FFFF
<b><i>Baud Rate</i></b>	9600

konfigurasi modul *XBee* pada *node coordinator* pada tabel 3.4

**Tabel 3.4 konfigurasi *node coordinator***

<b>PAN ID</b>	1105
<b><i>Destination Address High</i></b>	0
<b><i>Destination Address Low</i></b>	FFFF
<b><i>Baud Rate</i></b>	9600

b. Flowchart Pengiriman data topologi *tree*



**Gambar 3.6** Flowchart Pengiriman data topologi *tree*

### 3.6 SKENARIO PENGUJIAN SISTEM

Pada sub bab ini akan dijelaskan tentang alur pengiriman data untuk mengukur jaringan topologi *tree* pada masing – masing parameter performansinya.

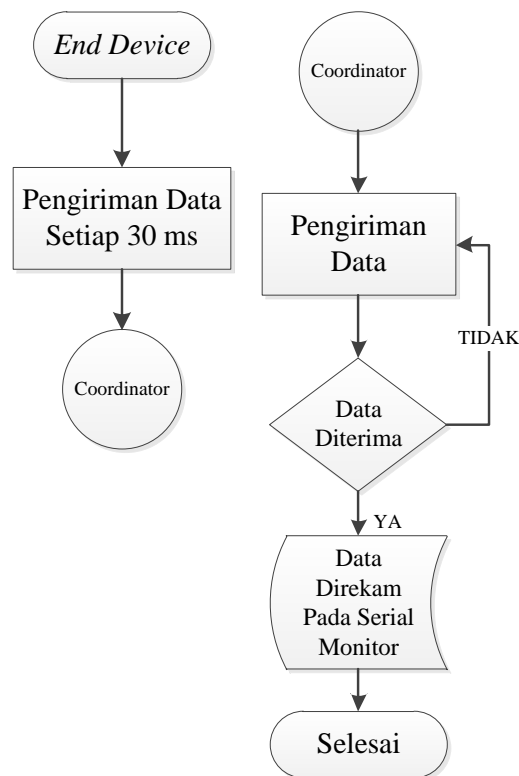
#### 3.6.1 Pengukuran jarak jangkauan Modul *XBee*

Tujuan dari pengukuran jangkauan jarak ini yaitu untuk mengetahui seberapa jauh jarak komunikasi yang dilakukan antara perangkat *XBee*. Data yang akan ditampilkan dalam pengukuran ini berupa *packet loss* dan RSSI. *Packet loss* adalah parameter untuk menguji berapa paket yang hilang selama proses pengiriman data. Dan RSSI merupakan ukuran daya yang diterima dalam komunikasi nirkabel. Perangkat akan mengirimkan beberapa variasi ukuran data mulai dari 10 *bytes*, 20 *bytes*, dan 30 *bytes*.

Data akan dikirimkan sebanyak 10 kali dan akan kembali lagi karena menggunakan sistem *loop back*. Data diukur dalam jarak kelipatan 5 meter, yaitu 5m, 10m, 15m, sampai tidak bisa terhubung. Hasil pengukuran berupa *packet loss* dan RSSI akan muncul pada *software* XCTU.

### 3.6.2 Pengukuran *Throughput*.

Tujuan dari pengukuran *throughput* ini untuk mengetahui seberapa besarnya data yang masuk pada satu waktu. Pada pengukuran *throughput* data yang dikirimkan bervariasi ukurannya. Data yang dikirimkan berasal dari *end device* yang kemudian akan diteruskan ke *coordinator*.



**Gambar 3.7 Flowchart Pengukuran *Throughput***

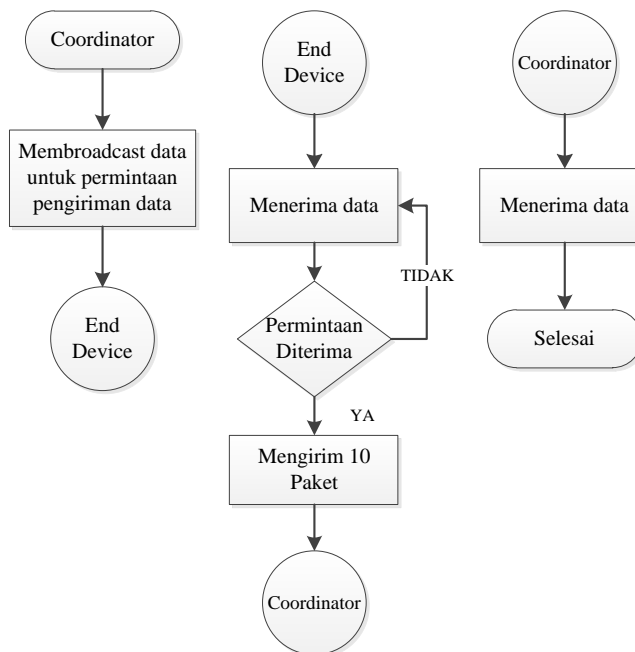
Seperti pada *flowchart* pada gambar 3.6 menjelaskan bahwa data akan dikirimkan dari *end device* dengan jeda 30 ms bertujuan untuk menghindari *overwhelmed* data yang dikirimkan dari *end device* dan untuk menghindarinya data tumpang tindih dengan *node* lain dalam ssatu jaringan. Selain itu pengukuran dibatasi setiap 10 detik unntuk memudahkann dalam peroses pengamatan



data yang masuk. Pada parameter throughput, sisi coordinator hanya bertugas menerima data dengan perekaman data menggunakan *software serial port monitor*. Jumlah paket yang masuk akan dihitung secara manual dan dikalkulasikan menggunakan rumus mencari nilai *throughput* sehingga akan didapatkan hasil akhir berupa nilai *throughput*.

### 3.6.3 Pengukuran Packet Loss

Pada pengukuran *packet loss* data dikirim pertama kali melalui *coordinator* seperti pada gambar 3.7 yang menunjukkan *flowchart* pada pengukuran *packet loss*. Tujuan dari pengukuran *packet loss* sendiri merupakan untuk mengukur seberapa besarnya data yang hilang pada proses transmisi data, serta untuk mengetahui pengaruh *collision* akibat pengiriman dari beberapa *node* menuju *node* yang sama secara bersamaan.

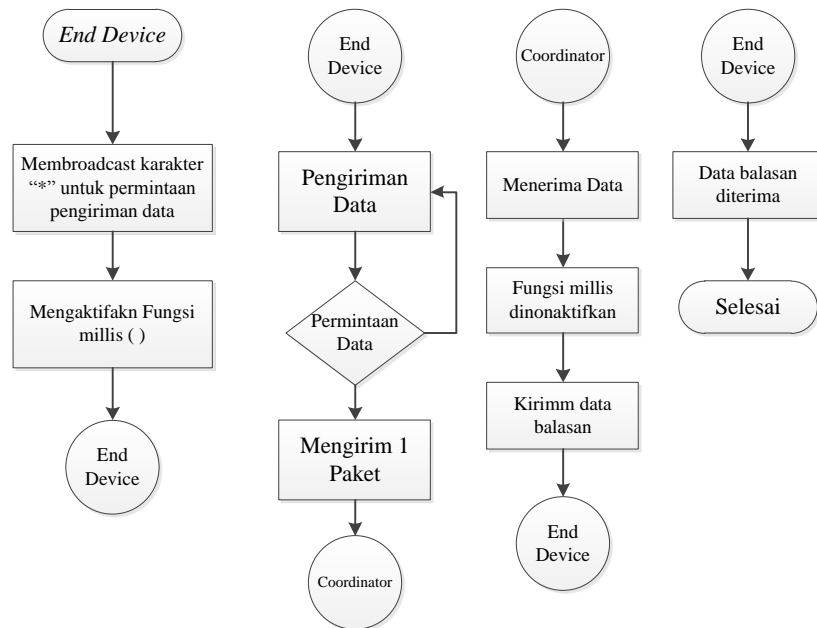


Gambar 3.8 Flowchart Pengukuran Packet Loss

### 3.6.4 Pengukuran Delay

Pengukuran *delay* sama seperti pada pengukuran *packet loss* yang menggunakan metode *coordinator* yang mengirim permintaan. Tujuan dari pengukuran *delay* adalah untuk

mengetahui waktu tuda atau lamanya proses transmisi pengiriman data dari satu *node* ke *node* lainnya



**Gambar 3.9 Flowchart Pengukuran Delay**

Pada pengukuran parameter *delay*, *coordinator* akan melakukan perminta dari pengiriman paket data *end device*. Ketika *coordinator* membroadcast permintaan, fungsi *millis* yang terdapat *microcontroller* Arduino diaktifkan. Fungsi *millis* akan terus berjalan menghitung waktu dalam satuan *mili second* sampai *end device* membalas permintaan dari *coordinator*. Ketika *end device* membalas data yang dikirimkan oleh *coordinator*, maka perhitungan waktu *delay* dapat dihitung dengan fungsi *millis*. Satuan nilai *delay* yang digunakan yaitu *millisecond*.