

BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

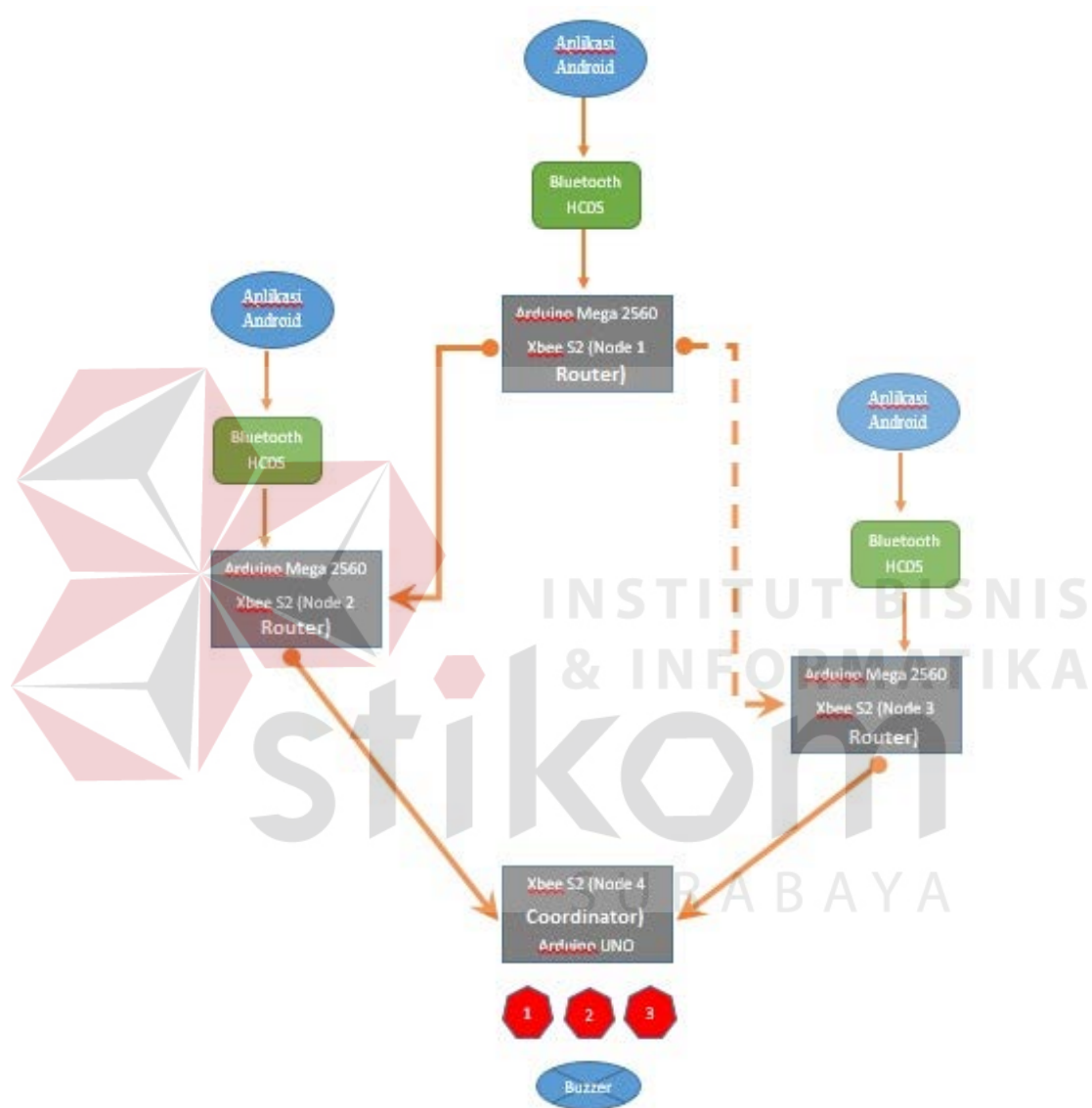
3.1 Metode Penelitian

Metode penelitian yang digunakan pada perancangan ini adalah komunikasi data multipoint *wireless sensor network*. Perancangan terdiri dari 4 buah *node* yang dikomunikasikan secara bersamaan dengan kecepatan data 115200bps. Dalam berkomunikasi, 4 *node* yang terdiri dari Xbee mengirim ke satu *base station* sehingga alamat *destination* pada masing - masing *node* nilainya sama dengan nomor seri Xbee bagian *base station*.

Untuk melakukan komunikasi *multipoint*, 4 buah *node* diset menjadi *router* (*router* 1, 2, 3 dan *Coordinator* 4), alamat *destination node* (DH (*Destination High*) dan DL (*Destination Low*)) adalah alamat *source coordinator* (SH (*Source High*) dan SL (*Source Low*)). Data yang diterima oleh *router*4 selanjutnya dipilah untuk mengetahui arah data dari *node* pengirim, dalam hal ini masing - masing *node* mengirimkan data sesuai isi data pada aplikasi android, dan data *limit switch*. Sehingga pemilahan data dapat diketahui dengan memeriksa data character yang masuk pada deretan protokol pengirim (*node*). Adapun data yang dikirim berupa nilai data dari sensor *limit switch* yang dipresentasikan dalam kondisi *normally open normally close* yang selanjutnya ditampilkan berupa nyala LED kedap kedip dan usara *buzzer*.

3.2 Model Perancangan

Pada perancangan ini penulis menggambarkan perancangan sistemnya seperti pada gambar 3.1 berikut.



Gambar 3.1 Penerapan Sistem dengan Topologi *Mesh*

Dari Gambar 3.1 didapatkan bahwa setiap *node* WSN memiliki tugas berbeda-beda seperti berikut:

- a) *Node Router/End Device 1*

Pada *node* ini, *node* bertanggung jawab mengirim perintah berupa notifikasi bantuan maupun rumah kosong dari aplikasi android melalui komunikasi serial *Bluetooth HC05*. Kemudian data bantuan maupun rumah kosong dan data kondisi pintu terbuka di terima oleh arduino mega 2560 selanjutnya data tersebut dikirimkan melalui komunikasi serial XBee S2 ke *node Router 2* atau *node Router 3* sesuai dengan protokol yang telah dibuat melalui modul arduino untuk di teruskan kembali ke *node Coordinator 4*.

b) *Node Router/End Device 2*

Pada *node* ini, *node* bertanggung jawab mengirim perintah berupa notifikasi bantuan maupun rumah kosong dari aplikasi android melalui komunikasi serial *Bluetooth HC05*. Kemudian data bantuan maupun rumah kosong dan data kondisi pintu terbuka di terima oleh arduino mega 2560 selanjutnya data tersebut dikirimkan melalui komunikasi serial XBee S2 ke *node Coordinator 4* sesuai dengan protokol yang telah dibuat melalui modul Arduino.

c) *Node Router/End Device 3*

Pada *node* ini, *node* bertanggung jawab mengirim perintah berupa notifikasi bantuan maupun rumah kosong dari aplikasi Android melalui komunikasi serial *Bluetooth HC05*. Kemudian data bantuan maupun rumah kosong dan data kondisi pintu terbuka di terima oleh arduino mega 2560 selanjutnya data tersebut dikirimkan melalui

komunikasi serial XBee S2 ke *node Router 4* sesuai dengan protokol yang telah dibuat melalui modul Arduino.

d) *Node Router/End Device 4*

Pada *node* ini, *node* bertanggung jawab atas penerima data yang telah dikirimkan oleh ketiga *node Router* (1, 2 dan 3). Data yang diterima oleh *node* ini kemudian diolah oleh arduino UNO untuk ditampilkan pada LED dan suara peringatan buzzer.

Terdapat 3 LED untuk menampilkan 3 *Node Router/End Device* (1 *node* 1 LED) dan 1 buzzer untuk memberikan suara peringatan. Data yang diterima sesuai dengan protokol pengiriman data. Pada *node Router/End Device 3* pengiriman data langsung dikirimkan ke *node Router/End Device 1* dan 2 sebagai perantara tanpa pengolahan data melalui modul arduino, hal ini dikarenakan proses pengolahan data dilakukan pada *node Coordinator* Penerima.

Terdapat 3 *Router/End Device* yang masing – masing *node Router 3* berkomunikasi secara *Broadcast* dengan *node Router/End Device 1* dan *node Router/End Device 2*. Pada *node Router/End Device 1* dan 2 berkomunikasi secara *unicast (point to point)* dengan *node Coordinator 4*. Pada *node Coordinator* penerima yang tersambung dengan *node Router/End Device* digunakan oleh pihak keamanan pada pos jaga untuk melihat hasil informasi yang dibutuhkan untuk meningkatkan sistem keamanan, dan data yang diperoleh valid (tidak tertukar). Dan dilakukannya pemisahan data dan pengelompokan data berdasarkan asal data seperti penjelasan berikut :

1. Data Bantuan ON (Tombol 1) :

- LED pin berkedip.
- Suara peringatan *buzzer* aktif

2. Data Bantuan OFF (Tombol 2) :

- LED pin mati.
- Suara peringatan *buzzer* berhenti

3. Data Rumah Kosong ON (Tombol 3) :

- LED pin menyala terus.

4. Data Rumah Kosong OFF (Tombol 4) :

- LED pin mati.

5. Pintu Terbuka (kondisi 1) :

- LED pin berkedip.
- Suara peringatan *buzzer* aktif

6. Data Rumah Kosong OFF

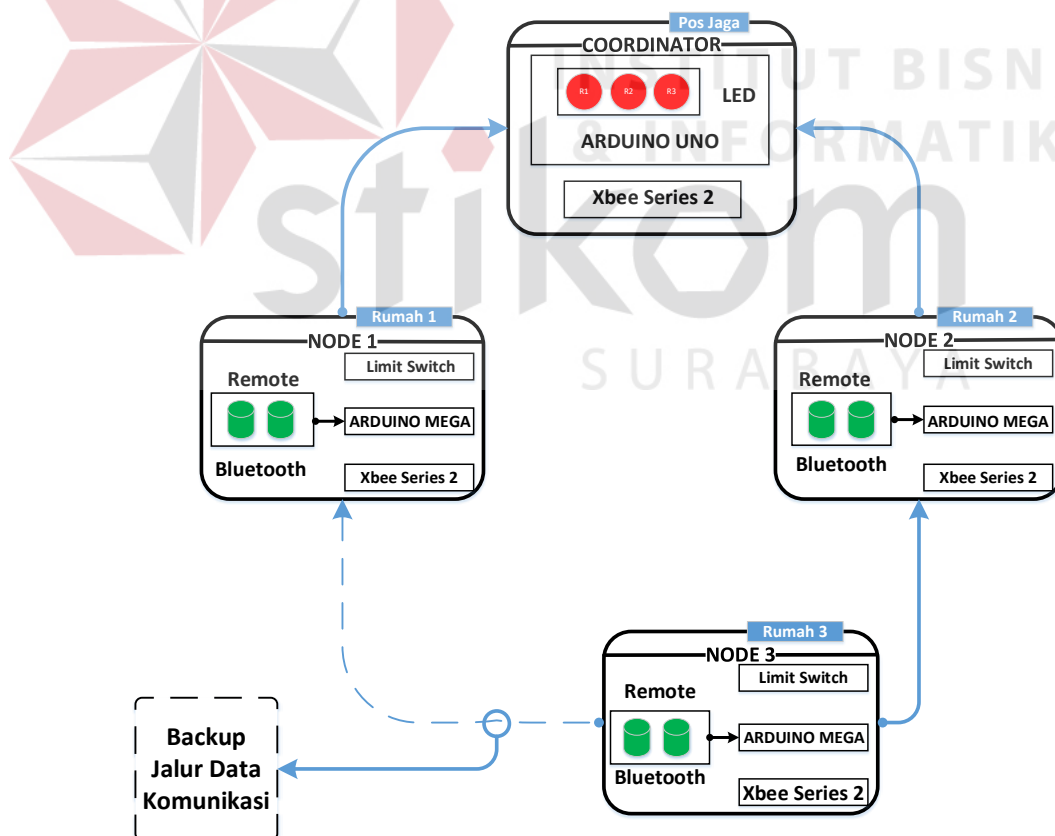
- LED pin mati.
- Suara peringatan *buzzer* berhenti

Hal ini dapat dilakukan dengan melihat isi data character yang sudah diberikan pada saat pengiriman data. Sedangkan penyimpanan data pada masing –

masing *node Router/End Device* pengirim yang terhubung dengan *node Coordinator* penerima digunakan untuk data pembanding antara data yang dikirim dengan data yang diterima apakah sudah terkirim dengan benar sesuai sistem yang telah di buat. Agar dapat diketahui data yang dikirim dari aplikasi android benar sesuai data yang di tampilkan melalui nyalanya LED ketika sistem ini dijalankan maupun suara peringatan buzzer.

3.3 Perancangan Sistem

Penelitian dilakukan dengan merancang komunikasi data yang terdiri dari beberapa *node*. *Node* dipasang sesuai dengan topologi *mesh* seperti yang ditunjukkan pada gambar berikut :



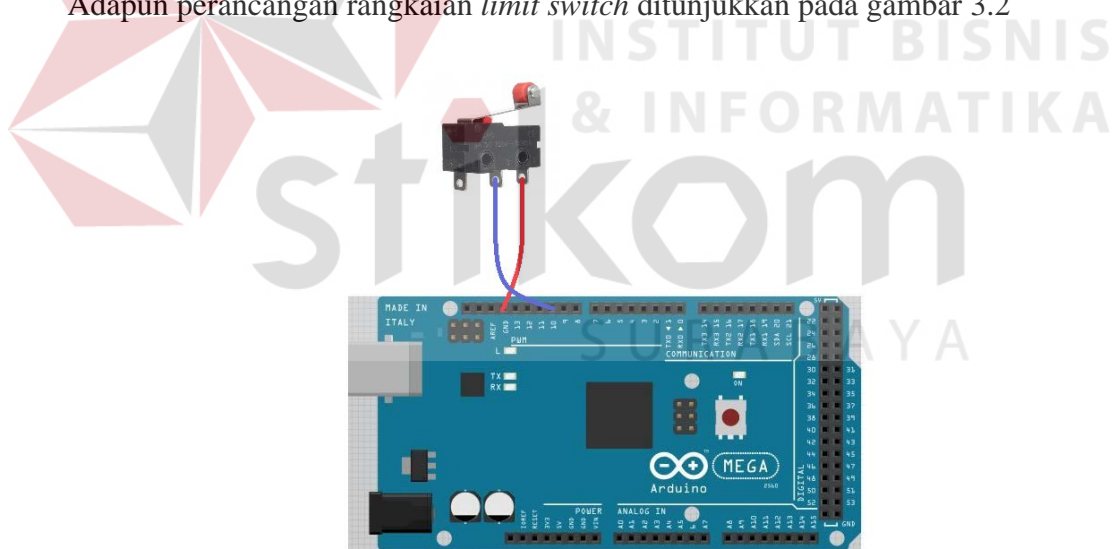
Gambar 3.2. Blok diagram penerapan sistem dengan topologi *Mesh*.

3.4 Perancangan perangkat keras

Selanjutnya penulis akan melakukan perancangan perangkat keras sesuai model perancangan sistem yang telah dibuat seperti perancangan alat keras yang saling terhubung dengan Arduino, Bluetooth HC05 dan XBee S2.

3.4.1 Perancangan Limit Switch

Untuk dapat mendeteksi keadaan pintu terbuka maupun pintu tertutup, maka dibutuhkan sensor. Sensor yang digunakan pada penelitian adalah *limit switch*. Sensor ini telah dilengkapi dengan pengkondisi *normally close* dan *normally open* yaitu kondisi 0 dan 1. dengan demikian keluaran dari sensor limit switch dapat langsung dibaca melalui pin digital pada modul Arduino Mega2560. Adapun perancangan rangkaian *limit switch* ditunjukkan pada gambar 3.2



Gambar 3.3 Hubungan Rangkaian Limit Switch dan Arduino

3.4.2 Perancangan Limit Switch dan Bluetooth HC05 Pada Pintu

Utama Rumah

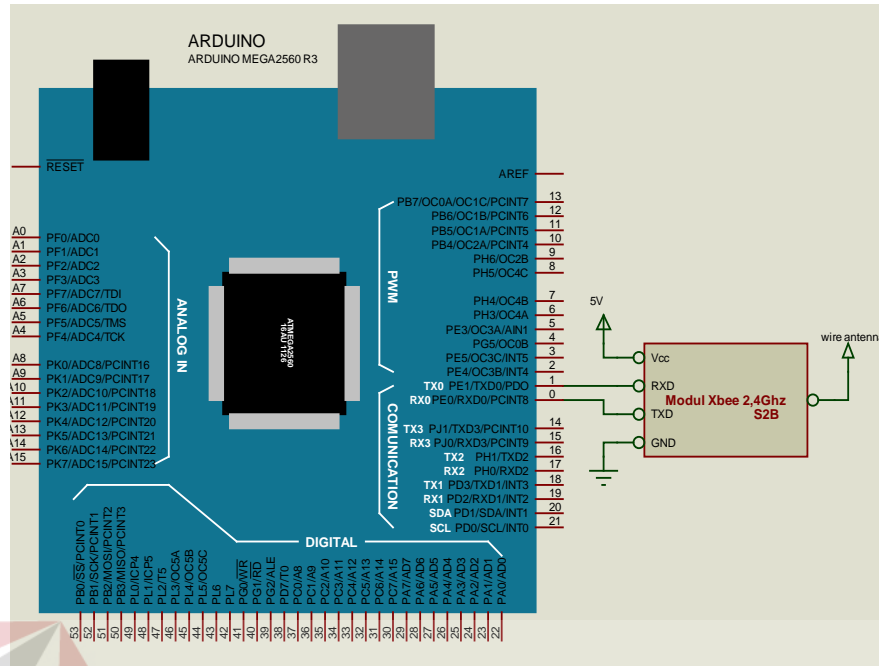
Untuk dapat mendeteksi keadaan pintu terbuka maupun pintu tertutup, maka dibutuhkan sensor. Sensor yang digunakan pada penelitian adalah *limit switch*. Posisi *limit switch* di pasang di atas pintu utama rumah. Adapun perancangan rangkaian *limit switch* ditunjukkan pada gambar 3.4



Gambar 3.4 Penempatan Limit Switch pada pintu rumah

3.4.3 Perancangan rangkaian Xbee Zigbee S2B

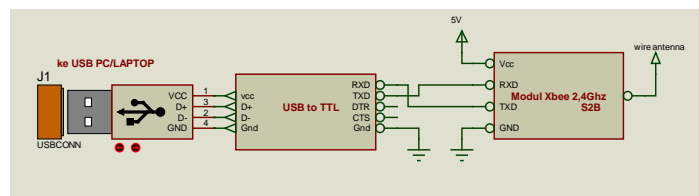
Agar modul arduino dapat berkomunikasi secara serial *wireless* dengan perangkat lain, maka dibutuhkan rangkaian *wireless* yang dalam perancangan ini menggunakan modul Zigbee S2B. modul zigbee dapat berkomunikasi *wireless* dan diakses menggunakan komunikasi serial TTL (*Time to Live*). Adapun port serial yang digunakan untuk pengendalian dan pembacaan modul Xbee adalah TX0 dan RX0 pada modul arduino sebagaimana ditunjukkan pada gambar 3.5:



Gambar 3.5 Hubungan Rangkaian Xbee dan arduino

3.4.4 Perancangan rangkaian USB to serial Xbee

Untuk dapat menerima data serial hasil pengiriman dari *node 1* dan *node 2* atau perangkat *wireless* Xbee pengirim, maka pada bagian penerima juga dibutuhkan Xbee. Sementara itu agar hasil pembacaan dan pengiriman data pada Xbee dapat diproses menggunakan laptop atau PC, maka dibutuhkan konverter USB to serial. Untuk itu pada perancangan ini digunakan modul USB to serial Xbee yang difungsikan khusus untuk menjembatani antarmuka UART antara komputer dengan Xbee. Adapun rangkaian modul USB to serial Xbee ditunjukkan pada gambar 3.6:



Gambar 3.6 Rangkaian modul USB to Serial Xbee pada *base station*

Sementara itu bentuk fisik dari modul USB to serial Xbee ditunjukkan pada gambar 3.6 :



Gambar 3.7 modul USB to serial Xbee

3.4.5 Perancangan LED dan *Buzzer* Pada Pos Jaga



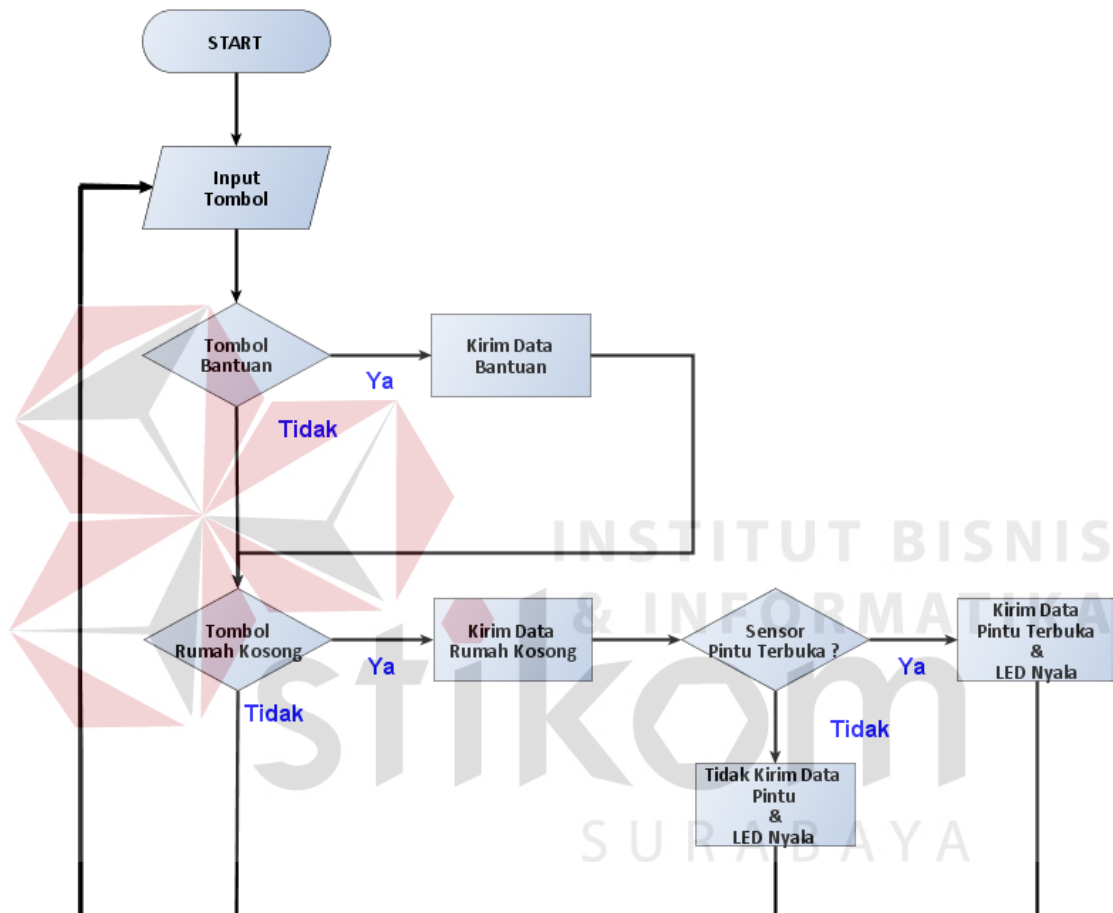
Gambar 3.8 Penempatan Lampu LED dan *Buzzer*

3.5 Perancangan Perangkat Lunak

Perancangan lunak sangatlah penting untuk menentukan alur jalannya program sesuai perancangan perangkat keras yang diimplementasikan pada tugas akhir ini. Perancangan perangkat lunak yang akan dilakukan yaitu perancangan perangkat lunak aplikasi android yang di fungsikan untuk penghuni rumah

memberikan informasi yaitu aplikasi *Bluetooth Controller*. Berikut adalah penjelasan masing – masing perancangan perangkat lunak :

3.5.1 Algoritma Penerapan Sistem Pada Rumah



Gambar 3.9 Flowchart Penerapan Sistem Pada Rumah.

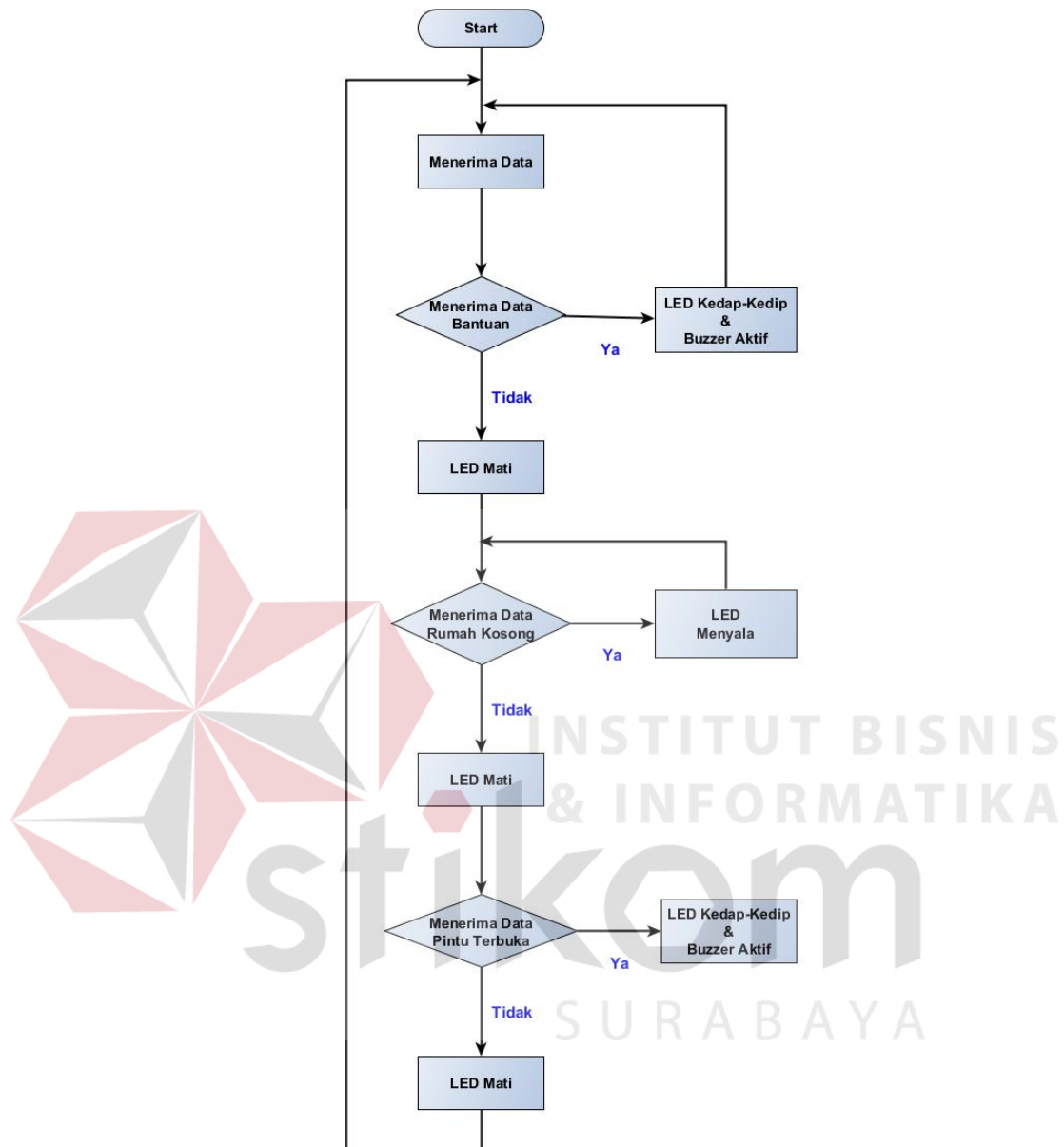
Seperti yang sudah dijelaskan diatas, hasil keluaran dari sensor kondisi pintu adalah berupa pintu terbuka atau pintu tertutup. Maka pada modul arduino dilakukan pembacaan melalui salah satu fungsi yang dimiliki oleh Arduino Mega2560, fungsi tersebut adalah *readDigital*. Pada pemrograman modul

Arduino Mega2560, *user* dimudahkan dengan beberapa fungsi yang sudah dimilikinya.

Selanjutnya data yang diperoleh akan ditransmisikan oleh pemancar sesuai dengan data yang telah diolah. Proses pengiriman data tidak langsung dikirim, karena data yang ada berupa character untuk pengidentifikasian asal data sesuai dengan asal *node router*.



3.5.2 Algoritma Penerapan Sistem Pada Pos Jaga



Gambar 3.10 Flowchart Penerapan Sistem Pada Pos Jaga

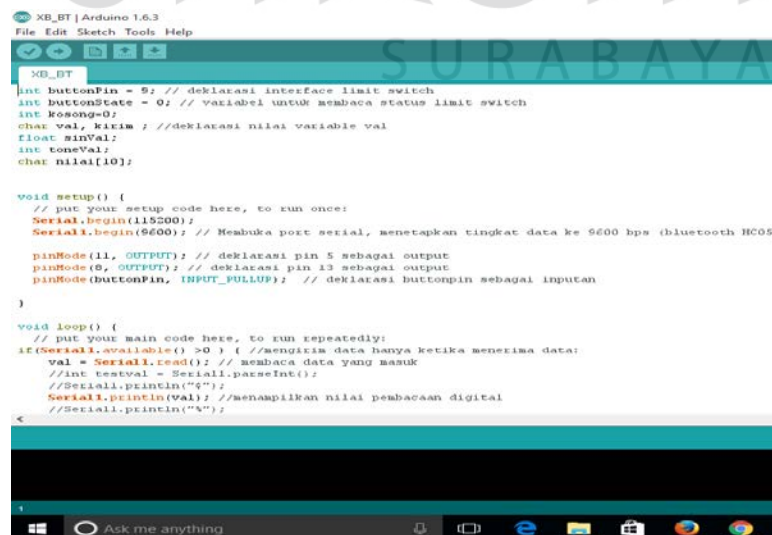
Pada dasarnya konsep dari pemberian LED dan buzzer adalah menerima setiap data, maka apabila ketika menerima data dari *node router* 1 maka data ditampilkan pada LED 1, secara langsung data *node* 1, 2 dan 3 akan ditransmisikan ke *node Coordinator*. Selain itu, karena pada transmisi data yang

didapat dari tiga *node*, maka untuk mempermudah dalam pengolahan data, maka dibuatlah algoritma seperti gambar 3.10.

Pemrograman Arduino Mega 2560 merupakan sebuah pemrograman modul, maka pemrograman langsung dilakukan pada setiap pin. Pin 9 digunakan sebagai Data input *limit Switch*. PIN 9 digunakan untuk memulai pengiriman data, *normally open* dan *normally Close*. Pengiriman dianggap selesai ketika *user* menekan tombol rumah kosong *off*.

Pengiriman data dilakukan sesuai dengan protokol yang sudah ditetapkan pada gambar 3.9, sehingga dapat disimpulkan bahwa pengiriman data berupa *string* atau *character*. Pengiriman data akan dipancarkan oleh modul pemancar yang sudah disediakan.

Pada tugas akhir ini, penulis menggunakan Arduino Mega2560 dan Arduino UNO sebagai mikrokontrolernya. *Software* yang digunakan untuk memprogram arduino tersebut ialah *software* Arduino IDE. Dan dari algoritma yang dibuat diatas maka dibuatlah program seperti gambar 3.9



```

XB_BT | Arduino 1.6.3
File Edit Sketch Tools Help

XB_BT
int buttonPin = 9; // deklarasi interface limit switch
int buttonState = 0; // variabel untuk membaca status limit switch
int kosong=0;
char val, Kirim ; //deklarasi nilai variable val
float sinVal;
int toneVal;
char nilai[10];

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  Serial.begin(9600); // Membuka port serial, menetapkan tingkat data ke 9600 bps (bluetooth HC05)

  pinMode(11, OUTPUT); // deklarasi pin 5 sebagai output
  pinMode(0, OUTPUT); // deklarasi pin 13 sebagai output
  pinMode(buttonPin, INPUT_PULLUP); // deklarasi buttonpin sebagai inputan
}

void loop() {
  // put your main code here, to run repeatedly:
  if(Serial.available() > 0) { //mengirim data hanya ketika menerima data:
    val = Serial.read(); // membaca data yang masuk
    //int testval = Serial.parseInt();
    //Serial.println("q");
    Serial.println(val); //menampilkan nilai pembacaan digital
    //Serial.println("v");
  }
}
  
```

Gambar 3.11 Tampilan program arduino pada *software* Arduino IDE

Berikut contoh pemrograman modul arduino Mega 2560 pada *node coordinator* yang diprogram pada Arduino IDE

a. Pembuatan variabel

Dalam pembuatan variabel, terdapat beberapa variabel yang digunakan oleh penulis seperti pada algoritma diatas. ini penulis menggunakan variabel tipe string yang bernama “string” untuk menampung data dari sensor, “nilai” untuk penampung data setelah dijadikan 10 bit. Pembuatan variabel ini diletakkan diluar fungsi void agar variabel ini dapat digunakan secara global. Berikut sebagai

contoh :

```
int buttonPin = 9; // deklarasi interface limit
switch
int buttonState = 0; // variabel untuk membaca status
limit switch
int kosong=0;
char val, val2 ; //deklarasi nilai variable val
float sinVal;
int toneVal;
char nilai[10];
int i=0;
```

b. fungsi void setup

Dalam fungsi void setup perintah akan dibaca 1 kali setelah program berjalan. Dalam tugas akhir ini penulis mengisi *baudrate* dan variabel - variabel dalam kondisi kosong, begitu juga pemberian nilai awal pada PIN – PIN yang digunakan. Berikut sebagai contoh :

```

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200); // port serial XBee
    Serial1.begin(9600); // port serial bluetooth
    pinMode(11, OUTPUT); // deklarasi pin 11 sebagai output
    pinMode(8, OUTPUT); // deklarasi pin 8 sebagai output
    pinMode(buttonPin, INPUT_PULLUP); //
}

```

c. fungsi void loop

Dalam void loop perintah akan dibaca berulang kali selama mikrokontroler tersambung dengan tegangan. Dalam tugas akhir ini penulis mengisi perintah bagaimana data diolah dan akhirnya dikirimkan, semua perintah ditulis pada void loop ini. Dan program ditulis sesuai dengan algoritma yang telah dibuat seperti pada gambar 3.10. Berikut contohnya :

```

void loop() {
    // put your main code here, to run repeatedly:
    if(Serial1.available() >0 ) { //mengirim data hanya ketika
    menerima data:
        val = Serial1.read(); // membaca data yang masuk
        Serial1.println(val); //menampilkan nilai pembacaan digital
    }
}

```

```

if(Serial.available() >0 ) { //mengirim data hanya ketika menerima
data:
    val2 = Serial.read(); // membaca data yang masuk
}

```

```

        if(val2=='F' || val2=='G' || val2=='H' || val2=='I')//
membatasi penerimaan data dari rumah 3

        Serial.println(val2);
    }

    if (val == '1') { // ketika mendeteksi kondisi bantuan led
berkedip dan buzzer mengeluarkan suara

        Serial.println(val);

        delay(1000);

        val='0';
    }

    if (val == '2') {

        Serial.println(val);

        delay(1000);

        val='0';
    }

    if (val == '3') {

        Serial.println(val);

        delay(1000);

        val='0';

        i=0;

        kosong=1;
    }

    if (val == '4') { // memberi perintah mematikan led pin 13 dan
mematikan suara buzzer kemudian mematikan led pin 6

        Serial.println(val);

        delay(1000);

        i=1;

        val='0';
    }

```

```

    }

    LSwitch();
}

int LSwitch()
{
    buttonState = digitalRead(buttonPin); //membaca nilai dari
buttonpin

    if(buttonState == 1 && kosong==1 && i==0){
        digitalWrite(11, HIGH);

        delay(200);

        //Serial.print('&');
        Serial.print('A');
        //Serial.println('#');
        kosong=0;//kembali posisi awal
    }
}

```

3.5.3 Pengambilan Data dari Aplikasi *Bluetooth Controller dan Limit*

Switch

Proses pengambilan data dilakukan saat semua alat terpasang, dan proses pengiriman data berlangsung. Seperti yang dijelaskan di atas, kita dapat melihat apakah sensor sudah berada pada posisi yang tepat.

Data pada masing – masing *router* pengirim akan di terima *router* penerima secara serial Xbee S2. Data berupa character yang diterima ditampilkan nyala lampu LED dan suara Buzzer menganalisa sistem dapat mentransmisikan data dari 3 *node* ke 1 titik *router* dengan benar.

List Input dan Output

Tabel 3.1 *List Input dan Output* Arduino Mega2560

<i>Node Pengirim</i>	Data Limit Switch	Pin Daya Power Limit Switch	Data LED	Data Suara <i>Buzzer</i>
Rumah 1	A	Pin 9, GND	1, 2, 3, 4	1, A
Rumah 2	B	Pin 9, GND	5, 6, 7, 8	5, B
Rumah 3	C	Pin 9, GND	F, G, H, I	F, C

