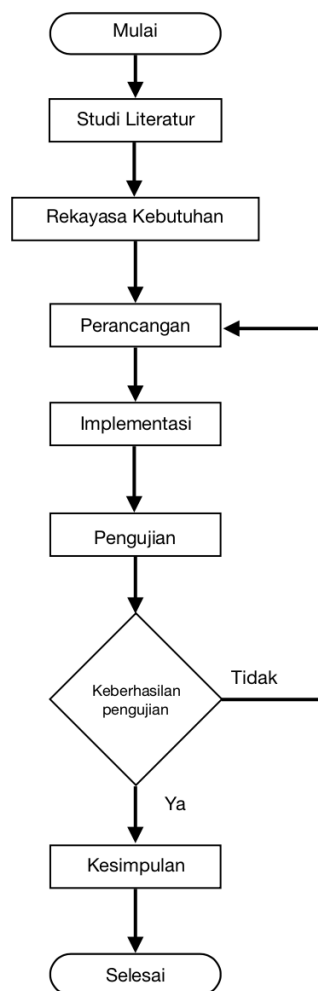


## BAB 3 METODOLOGI

Pada bab ini memberikan penjelasan mengenai metodologi atau langkah-langkah yang digunakan dalam penelitian ini. Penelitian ini bersifat implementatif yang mengimplementasikan sistem pengenalan layanan dan perangkat sensor dan aktuator berbasis arsitektur *publish-subscribe* menggunakan protokol mqtt pada objek rumah cerdas. Penentuan alur metode penelitian sebagai langkah yang ditempuh untuk menyelesaikan penelitian ini secara sistematis. Gambar 3.1 merupakan diagram alir yang berisi tahap-tahapan yang dilakukan dalam pembuatan sistem ini yang meliputi studi kepustakaan, analisis kebutuhan, perancangan, implementasi, pengujian dan analisis serta pengambilan keputusan.



Gambar 3.1 Diagram alir metode penelitian

### 3.1 Studi Kepustakaan

Studi kepustakaan merupakan tahapan pencarian kepustakaan dan melakukan penyusunan teori dasar dan referensi yang akan menunjang penelitian ini. Studi kepustakaan pada penelitian ini dilakukan dengan

pemahaman terhadap tinjauan pustaka dan dasar teori yang dapat diperoleh dari buku, jurnal, skripsi dan website terkait. Tinjauan pustaka dan dasar teori tersebut meliputi :

1. Arsitektur *Publish-Subscribe*

Salah satu arsitektur modern yang saat ini banyak dikembangkan untuk kebutuhan perangkat cerdas serta arsitektur ini memungkinkan untuk pengiriman secara realtime antar beberapa *device*.

2. MQTT

*Message Queuing Telemetry Transport* merupakan salah satu protokol yang menerapkan arsitektur *publish-subscribe* dan didesain untuk kebutuhan komunikasi pada perangkat *Internet of Things*.

3. Sistem Pervasif

Suatu metode yang mempelajari tentang bagaimana mengidentifikasi serta mengenali suatu layanan dan perangkat sensor maupun aktuator pada suatu jaringan *Wi-Fi*. Setelah mengidentifikasi kemudian layanan dan perangkat tersebut langsung dapat berinteraksi dengan penggunanya.

4. *NodeMCU*

Sebuah mikrokontroler yang telah dilengkapi konektivitas *wireless* menggunakan *Wi-Fi* yang dapat digunakan untuk mengakuisisi setiap data sensor yang digunakan serta memilih spesifikasi pemrosesan data lebih cepat dibandingkan Arduino uno.

5. JSON (*JavaScript Object Notation*)

Salah satu format teks yang umum dipakai untuk kebutuhan pertukaran informasi baik itu antar *device* maupun di sisi *server*. Format teks ini juga telah didukung hampir semua bahasa pemrograman.

## 3.2 Analisa Kebutuhan

Analisis kebutuhan ditujukan untuk menganalisa mengenai kebutuhan yang akan mendukung dalam proses penelitian ini. Analisa kebutuhan pada penelitian ini adalah sebagai berikut.

### 3.2.1 Kebutuhan Perangkat Keras

1. Laptop
2. *NodeMCU*
3. Sensor PIR dan LDR
4. Lampu
5. *Relay*
6. Kabel Jumper

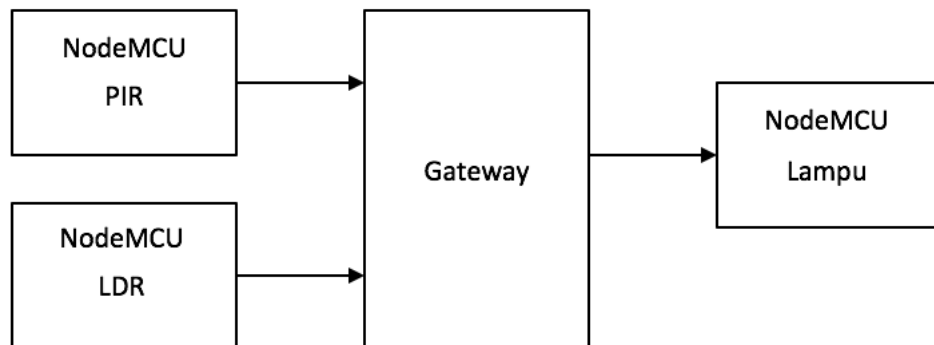
### 3.2.2 Kebutuhan Perangkat Lunak

1. Arduino IDE
2. Python
3. *ArduinoJson Library*
4. *Mqtt Library*

## 5. Mosquitto MQTT *Broker*

### 3.3 Perancangan Sistem

Perancangan sistem dilakukan setelah melewati tahap analisis kebutuhan sistem yang harus terpenuhi terlebih dahulu baik dari sisi perangkat keras dan perangkat lunak. Tujuan perancangan sistem agar tahapan implementasi berjalan secara sistematis dan terstruktur. Perancangan sistem pada penelitian ini digambarkan pada diagram blok pada Gambar 3.2.



**Gambar 3.2 Blok diagram sistem**

Berdasarkan Gambar 3.2 perancangan sistem dapat dijelaskan pada poin-poin berikut dibawah ini :

1. Masing-masing *node* sensor tersebut menggunakan papan mikrokontroler NodeMCU yang akan mengakuisisi data setiap sensornya yakni sensor PIR dan LDR, kemudian mengirimkan data yang telah dibungkus kedalam format teks JSON ke *gateway* yang berjalan dengan memanfaatkan aplikasi Mosquitto MQTT *Broker* menggunakan komunikasi *Wi-Fi* serta menggunakan protokol komunikasi MQTT. Dimana *node* sensor akan berperan sebagai *publisher*.
2. Perangkat *gateway* yang berjalan dengan memanfaatkan aplikasi Mosquitto MQTT *Broker* akan mendapatkan keseluruhan akuisisi data sensor dalam format teks JSON dalam rangka mengenali perangkat dan layanan baru dari sensor maupun aktuator. Selain itu *gateway* akan bertindak sebagai pengatur untuk menentukan relasi terhadap sensor dan aktuator dan kemudian bertindak sebagai pusat informasi yang dikirimkan dari berbagai sensor dan aktuator.
3. *Node* aktuator menggunakan papan mikrokontroler NodeMCU yang akan memproses setiap data yang telah diteruskan oleh *gateway*, sehingga *node* aktuator dapat mengambil suatu keputusan untuk menyalakan atau mematikan lampu serta mengirimkan kondisi terakhir ke *gateway* untuk kembali diolah.

### 3.4 Implementasi

Pada tahap implementasi akan mengacu pada hasil perancangan yang telah dilakukan. Tahapan implementasi memiliki 2 tahapan yang harus dilakukan yakni meliputi implementasi perangkat lunak dan perangkat keras.

#### 3.4.1 Implementasi Perangkat Keras

Pada tahapan implementasi perangkat keras memiliki beberapa tahapan yakni implementasi perangkat keras pada *node* sensor serta *node* aktuator. Setelah itu keseluruhan tahapan dibentuk agar dapat terhubung antar satu sama lain baik sensor maupun aktuator.

##### 1. Implementasi perangkat keras *node* sensor

Implementasi perangkat keras untuk *node* sensor menggunakan sensor LDR dan sensor PIR. Masing-masing sensor terhubung dengan mikrokontroler nodemcu dengan menggunakan komunikasi *Wi-Fi*. Sensor LDR yang dihubungkan ke mikrokontroler nodemcu sebagai sensor pendeteksi tingkat intensitas cahaya. Sensor PIR yang dihubungkan ke mikrokontroler nodemcu sebagai sensor pendeteksi keberadaan manusia pada suatu ruangan.

##### 2. Implementasi perangkat keras *node* aktuator

Implementasi perangkat keras untuk *node* aktuator menggunakan lampu rumah serta *relay*. Masing-masing *device* tersebut terhubung dengan mikrokontroler nodemcu dengan menggunakan komunikasi *Wi-Fi*. *Relay* yang dihubungkan ke mikrokontroler nodemcu bertindak sebagai switch digital untuk memicu lampu rumah tersebut.

#### 3.4.2 Implementasi Perangkat Lunak

Pada tahapan implementasi perangkat lunak meliputi implementasi perangkat lunak pada *gateway* dan *node* sensor maupun *node* aktuator.

##### 1. Implementasi perangkat *gateway*

Implementasi perangkat *gateway* dimulai dengan menerapkan mekanisme untuk membuat sistem secara pervasif dapat mengenali suatu perangkat baru serta membuat relasi antar perangkat.

##### 2. Implementasi *node* sensor dan aktuator

Implementasi *node* sensor dan aktuator dengan mengimplementasikan kode program untuk mengakuisisi data sensor dan aktuator serta mengintegrasikan *node* terhadap sistem pervasif pada *gateway*

### 3.5 Pengujian dan Analisis

Tahapan pengujian dan analisis dilakukan untuk mengetahui kesesuaian sistem setelah proses implementasi yang dibandingkan dengan spesifikasi sistem pada saat tahap perancangan. Proses pengujian yang dilakukan yakni sebagai berikut :

1. Pengujian dengan menghubungkan perangkat *gateway*, sensor dan aktuator terhadap jaringan *wifi* dan *broker*.
2. Pengujian dengan melakukan inialisasi objek *metadata* pada perangkat sensor dan aktuator.
3. Pengujian dengan melakukan pengenalan perangkat dan layanan baru dari perangkat aktuator.
4. Pengujian dengan melakukan pembuatan relasi antara sensor dengan aktuator.
5. Pengujian dengan melakukan pengiriman data dari perangkat aktuator dan sensor serta melakukan pengambilan keputusan pada perangkat aktuator.
6. Pengujian dengan melakukan penghapusan relasi sensor terhadap perangkat aktuator serta *gateway*.

Setelah tahapan pengujian ini yang selanjutnya akan dilakukan tahap analisis untuk mengetahui tingkat kesesuaian dengan perancangan dan spesifikasi yang telah dibuat.

### **3.6 Kesimpulan**

Pengambilan kesimpulan dilakukan setelah semua tahapan sudah berjalan yang dimulai dari tahap perancangan, implementasi, pengujian dan analisis. Kesimpulan diambil dari tahap pengujian dan analisis. Kesimpulan diambil untuk mengetahui informasi fungsionalitas secara keseluruhan dari semua sub-sistem baik dari sisi sensor, aktuator dan *gateway*. Pada tahap ini juga terdapat saran untuk pengembangan sistem yang lebih sempurna baik dari kekurangan sistem ini maupun ide tambahan untuk mengembangkan sistem ini.