

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Penelitian Terkait

Beberapa Penelitian telah dilakukan dengan penelitian terkait, diantaranya sebagai berikut:

Penelitian yang berjudul “*Prototype Smart Home Dengan Modul NodeMCUESP8266 Berbasis Internet of Things (Iot)*”. Aplikasi dari sistem *Smart Home* berbasis (IoT) *Internet of Things* ini menggunakan modul *NodeMCUESP8266* sebagai *microcontroller* dan aplikasi android *Blynk* sebagai alat pengendali ataupun monitoring. Sistem ini mengedalikan beberapa alat seperti lampu, kipas angin, selain itu juga dapat mengawasi keadaan seperti monitoring suhu ruangan, pendeteksi pergerakan di suatu ruangan, dan pendeteksi kebocoran gas. (Hidayati, Nurul, Dkk. 2018)[3].

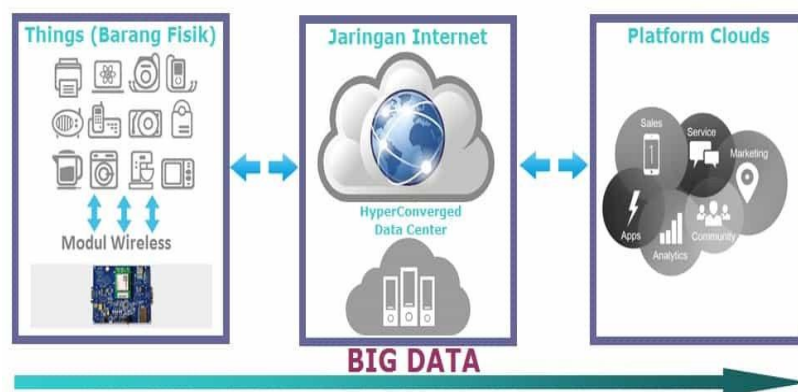
Penelitian dengan judul “Perancangan pengendali rumah menggunakan *smartphone* android dengan konektivitas *bluetooth*”. Penelitian ini dilakukan untuk pengujian fungsional sistem dengan parameter agar dapat mengeksekusi perintah dari *smartphone* android, pemeriksaan timer berjalan sesuai waktu masukan dari user dan sensor dapat membaca kondisi lampu dengan benar. Hasil dari pengujian fungsional yaitu komunikasi *wireless* tetap dapat dilakukan di dalam ruangan yang terdapat tembok dan jarak 20 meter dalam ruang terbuka. Fungsional timer berjalan dengan benar sesuai dengan nilai masukan user dan membaca kondisi lampu oleh sensor sesuai

dengan hasil yang diinginkan. (Bayu sadewo, A. D., Widasari, E. R., & Muttaqin, A. (2017)[4].

Penelitian dengan judul “Implementasi *Smart Home* Menggunakan Bot Telegram Sebagai Kontroller”. *Prototype* dari *Smart Home* dengan sistem Bot Telegram sebagai kontrolernya dengan parameter pengujian yang digunakan adalah *delay* dan *throughput*. Bot telegram dapat mengontrol pintu dan lampu pada miniatur rumah. Metode pengontrolan *Smart Home* ini diharapkan dapat berguna ketika ingin menyalakan atau mematikan lampu, dan ingin membuka ataupun menutup pintu. (Sitinjak, Riwanto, Dkk. 2020)[5].

## 2.2. Landasan Teori

### 2.2.1. *Internet of Things* (IoT)



Gambar 2. 1 Konsep IoT

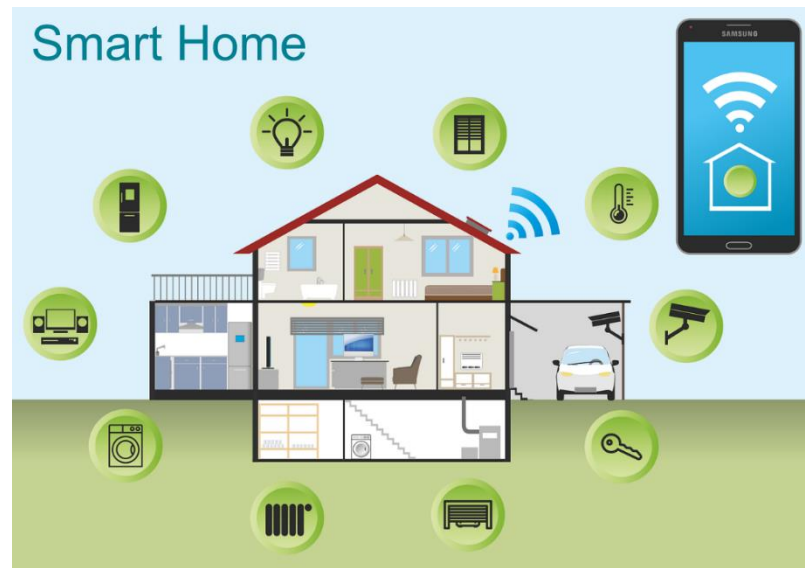
*Internet of Things* (IoT) didefinisikan sebagai sebuah infrastruktur koneksi jaringan global yang menghubungkan benda fisik dan virtual melalui pengumpulan data dan teknologi komunikasi, menurut *Coordinator and Support Action for Global RFID-related Activities and Standardization*. Infrastruktur *Internet of Things* terdiri

dari internet sebelum pengembangannya serta jaringan yang sudah ada. Hal ini memungkinkan identifikasi objek, identifikasi sensor, dan kemampuan koneksi, yang sangat penting untuk pengembangan layanan dan aplikasi kooperatif yang berdiri sendiri. Ini juga dicirikan dengan tingkat otonomi pengumpulan data yang tinggi, transfer peristiwa, konektivitas jaringan, dan interoperabilitas. *Internet of Things* (IoT) didefinisikan oleh IEEE (*Institute of Electrical and Electronics Engineers*) sebagai sebuah jaringan di mana setiap benda memiliki sensor yang terhubung ke jaringan internet[6].

Konsep "*Internet of Things*" terdiri dari tiga komponen utama: benda fisik atau nyata yang telah diintegrasikan ke dalam modul sensor, koneksi internet, dan pusat data pada server yang digunakan untuk menyimpan data dan informasi dari aplikasi. Data yang dikumpulkan menjadi "*big data*" kemudian akan diproses dan dianalisis oleh lembaga pemerintah, perusahaan terkait, dan lembaga lain untuk tujuan yang berbeda.

### **2.2.2. *Smart Home***

*Smart Home* adalah aplikasi gabungan antara teknologi dan layanan yang dirancang untuk membuat rumah lebih aman, efisien, dan nyaman. Rumah pintar biasanya terdiri dari otomatis, perangkat monitoring, dan perangkat kontrol. Selain dapat bekerja otomatis dan dikontrol dari jauh, rumah pintar juga dapat menghemat energi, waktu, dan uang[7].



Gambar 2. 2 *SmartHome*

### 2.2.3. *NodeMCUESP8266*

*NodeMCU* adalah board elektronik yang berbasis chip *ESP8266* yang memiliki kemampuan untuk menjalankan fungsi mikrokontroler serta memiliki kemampuan untuk terhubung ke internet melalui WiFi. *NodeMCUESP8266* memiliki beberapa pin I/O, yang memungkinkan pengembangan aplikasi pengawasan dan kontrol untuk proyek IoT. Bentuk fisiknya memiliki port USB (*mini USB*), yang memudahkan pemrograman[8].

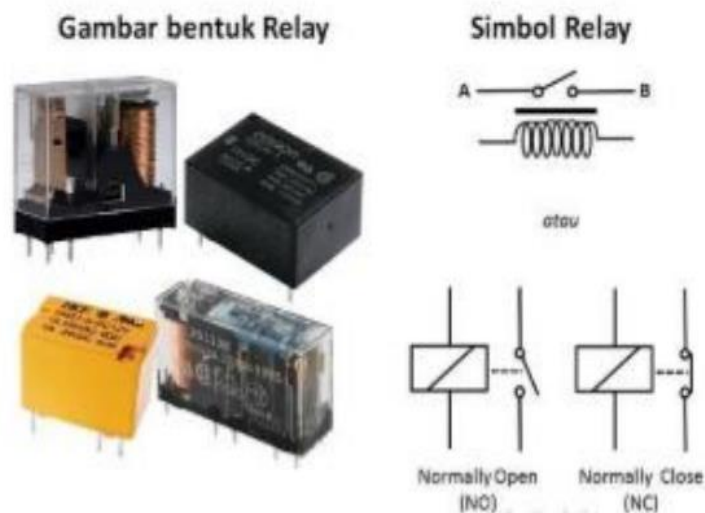


Gambar 2. 3 *NodeMCUESP8266*

*NodeMCUESP8266* adalah modul turunan pengembangan dari keluarga modul IoT (*Internet of Things*) *ESP8266* tipe ESP-12. Fungsinya hampir sama dengan platform modul *Arduino*, tetapi dikhususkan untuk “*Connected to Internet*”.

Modul Wi-fi *NodeMCU* ialah *firmware* interaktif berbasis LUA Espressif *ESP8266* Wifi SoC. *NodeMCUESP8266* v0.9 memiliki 4MB *flash*, 11 pin GPIO dimana 10 diantaranya dapat digunakan untuk PWM, 1 pin ADC, 2 pasang UART, Wifi 2,4GHz serta mendukung WPA/WPA2.

#### 2.2.4. Modul *Relay*



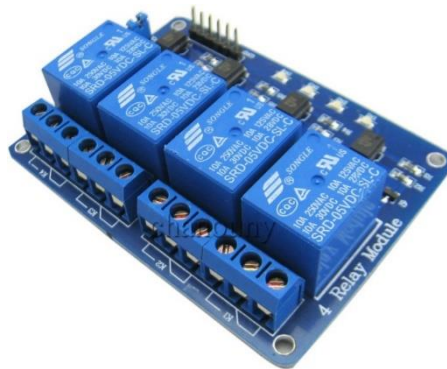
Gambar 2. 4 *Relay*

*Relay* merupakan saklar elektronik yang dapat membuka atau menutup rangkaian dengan menggunakan control dari rangkaian elektronik lain. Sebuah *Relay* tersusun atas kumparan, pegas, saklar (terhubung pada pegas) dan 2 kontak elektronik[9].

Berdasarkan dari prinsip dasar cara kerjanya, *Relay* beroperasi dengan menggunakan medan magnet untuk menggerakkan saklar. Lalu ketika kumparan *Relay* diberikan tegangan, medan magnet yang dihasilkan oleh kumparan elektromagnetik akan menarik saklar dari kontak NC (*Normally Closed*) ke kontak NO (*Normally Open*). Jika tegangan pada kumparan dimatikan, medan magnet akan hilang, dan pegas akan menarik saklar dan kembali ke kontak NC.

Gambar 2.4 menjelaskan bentuk komponen *Relay* yang terdiri dari 4 macam komponen dasar yaitu :

1. Koil
2. *Armature*
3. *Contact Point* (Saklar)
4. Per



Gambar 2. 5 Modul *Relay*

Gambar 2.5 merupakan gambar modul *Relay* 4 chanel yang sudah dilengkapi dengan komponen *optocoupler* atau *optoisolator* kapasitas maksimum setiap *Relay* adalah 10 A.

Rangkaian penggerak dari *Relay* menggunakan komponen *optocoupler*. Sebagai pengaman karena *optocoupler* bekerja berdasarkan sinar infra merah yang terdeteksi oleh komponen *phototransistor* yang menyebabkan terjadinya hubungan atau saklar on dan arus listrik dan tegangan yang lebih tinggi pada beban. Tegangan dan arus ini tidak akan masuk ke bagian mikrokontroler sehingga alat menjadi aman.

#### 2.2.5. Aplikasi *Blynk*



Gambar 2. 6 Simbol Aplikasi *Blynk*

*Blynk* adalah platform untuk aplikasi OS *Mobile* (iOS dan Android) yang dirancang untuk mengendalikan modul seperti *Arduino*, *Raspberry Pi*, *ESP8266*, *WEMOS D1*, dan modul sejenisnya melalui Internet. Aplikasi ini memungkinkan pengguna untuk membuat antarmuka grafis untuk proyek mereka dengan mudah menggunakan metode *drag and drop Widget*. Penggunaannya sangat sederhana dan memungkinkan pengaturan dalam waktu kurang dari 5 menit. *Blynk* tidak terbatas pada papan atau modul tertentu. Melalui platform ini, pengguna dapat mengontrol apa saja dari jarak jauh, kapan saja, asalkan terhubung dengan internet yang stabil. Inilah yang disebut dengan sistem *Internet of Things (IoT)*[10].

### 2.2.6. *Arduino IDE*

*Arduino* adalah platform *open-source* yang populer untuk membuat prototipe elektronik dan aplikasi IoT. *Arduino* dapat diprogram menggunakan bahasa pemrograman khusus melalui *Arduino IDE (Integrated Development Environment)*. *Arduino IDE* adalah perangkat lunak yang digunakan untuk mengembangkan dan memprogram mikrokontroler *Arduino* [11].

*Arduino IDE* adalah perangkat lunak *open-source* yang digunakan untuk memprogram mikrokontroler *Arduino*. Tersedia untuk *Windows, Mac OS, dan Linux*, *Arduino IDE* dapat diunduh secara gratis dari situs resmi *Arduino*. Memanfaatkan bahasa pemrograman C atau C++, *Arduino IDE* memungkinkan pengguna untuk membuat program untuk mikrokontroler *Arduino*. Antarmuka yang sederhana dan mudah digunakan membuatnya cocok bagi pengguna pemula dalam pemrograman.



Gambar 2. 7 Simbol *Arduino IDE*



*Arduino* IDE memiliki banyak kelebihan yang membuatnya sangat populer di kalangan pengembang IoT. Beberapa kelebihan tersebut antara lain:

1. *Open-source*: *Arduino* IDE adalah perangkat lunak *open-source* yang dapat diakses dan digunakan secara gratis oleh siapa saja.
2. Mudah digunakan: *Arduino* IDE memiliki antarmuka yang sederhana dan mudah digunakan, bahkan untuk pengguna yang belum berpengalaman dalam pemrograman.
3. Kompatibilitas yang baik: *Arduino* IDE kompatibel dengan berbagai jenis board *Arduino* dan mudah diintegrasikan dengan berbagai komponen elektronik.
4. *Library* dan contoh program yang lengkap: *Arduino* IDE menyediakan banyak *Library* dan contoh program yang dapat digunakan untuk mempermudah pembuatan program.

### **2.2.7. Flowchart**

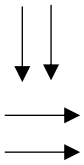
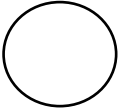
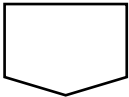



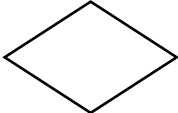
*Flowchart* atau juga dikenal sebagai bagan alur, adalah diagram yang menunjukkan langkah-langkah dan keputusan yang diambil untuk menjalankan proses suatu program. Setiap langkah digambarkan dalam bentuk diagram dan dihubungkan dengan garis atau arah panah [12].



Salah satu fungsi utama *Flowchart* adalah untuk memberi gambaran alur sebuah program dari satu proses ke proses lainnya sehingga alur program menjadi mudah dipahami oleh semua orang.

Selain itu, *Flowchart* juga berfungsi untuk menyederhanakan rangkaian prosedur sehingga lebih mudah untuk memahami informasi tersebut.

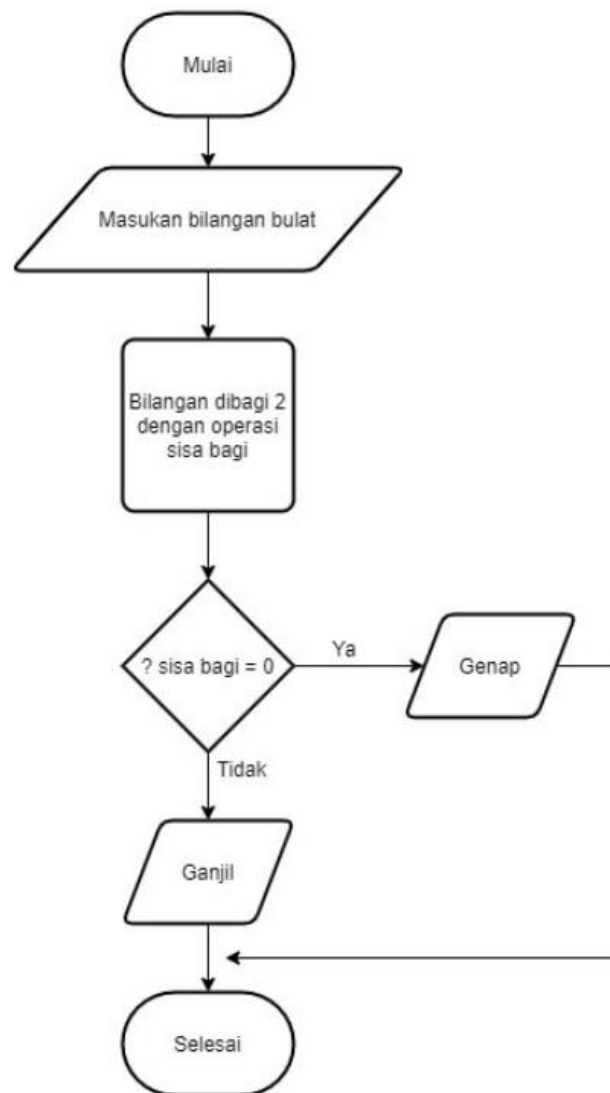
Pada dasarnya simbol-simbol dalam *Flowchart* memiliki arti yang berbeda-beda. Berikut adalah simbol-simbol yang sering digunakan dalam proses pembuatan *Flowchart*:

Tabel 2.1 Penjelasan *Flowchart* Program

No.	Simbol	Penjelasan
1.		<b>Flow</b> Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.
2.		<b>On-Page Reference</b> Simbol untuk keluar – masuk atau penyambungan proses dalam lembar kerja yang sama.
3.		<b>Off-Page Reference</b> Simbol untuk keluar – masuk atau penyambungan proses dalam lembar kerja yang berbeda.
4.		<b>Terminator</b> Simbol yang menyatakan awal atau akhir suatu program.
5.		<b>Process</b> Simbol yang menyatakan suatu proses yang dilakukan komputer.
6.		<b>Input/output</b> Simbol yang menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung peralatan.
7.		<b>Decision</b> Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.

No.	Simbol	Penjelasan
8.		<b>Predefine Process</b> Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
9.		<b>Preparation</b> Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Simbol-simbol di atas memiliki jenis dan fungsi yang berbeda-beda. Ada yang berfungsi untuk menghubungkan satu simbol dengan simbol lainnya seperti simbol *flow*, *on-page* dan *off-page reference*. Selain itu ada juga simbol yang berfungsi untuk menunjukan suatu proses yang sedang berjalan, dan yang terakhir terdapat simbol yang berfungsi untuk memasukan *input* dan menampilkan *output*. Berikut adalah contoh dalam penggunaan *Flowchart*:



Gambar 2. 8 Contoh Penggunaan *Flowchart*

Gambar 2.8 merupakan sebuah contoh Flowchart sederhana untuk menentukan apakah bilangan yang dimasukan ganjil atau genap.