

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Terdahulu**

Penelitian terdahulu menjadi salah satu acuan penulis dalam pembuatan laporan akhir ini, sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian membuka pintu menggunakan teknologi *wireless*.

Rujukan penelitian yang pertama yaitu jurnal Arafat, S.Kom, M.Kom dengan judul Sistem Pengamanan Pintu Rumah Berbasis *Internet of Things* (IoT) dengan ESP8266. Dalam penelitiannya peneliti menggunakan aplikasi *Blynk* pada *smartphone* agar alat dapat secara otomatis membuka pintu dan menutup pintu. Namun keadaan pintu akan mengunci saat tidak ada tegangan listrik yang disebabkan oleh pemadaman listrik, tapi pintu dapat di buka dan di tutup secara manual menggunakan *switch* pada setiap pintu. Adapun gagalnya perintah atau notifikasi sering disebabkan karena jaringan internet yang lambat.

Rujukan penelitian yang kedua yaitu jurnal Ade Septryanti dan Fitriyani dengan judul Rancang Bangun Aplikasi Kunci Pintu Otomatis Berbasis Mikrokontroler Arduino menggunakan *Smartphone* Android. Dalam penelitiannya komunikasi antara *smartphone* android dengan mikrokontroler dilakukan secara *wireless*, dan penguncian tidak akan bekerja jika *smartphone* android diluar jarak jangkauan pancaran *wireless* dari *bluetoothmodule* karena sambungan *bluetooth* akan terputus secara otomatis. Agar dapat membuka kunci pintu diperlukan gambar berupa QRCode yang telah di program secara khusus pada mikrokontroler yang teridentifikasi antara aplikasi dengan mikrokontroler.

Sedangkan penelitian yang akan dilakukan oleh peneliti saat ini tidak jauh berbeda dengan penelitian sebelumnya yaitu untuk membuka pintu menggunakan teknologi *wireless* atau *Internet of Things* (IoT) yang berbasis mikrokontroler *Wemos*.

## 2.2 *Wireless*

Wireless merupakan sebuah jaringan untuk menghubungkan jaringan internet dan komputer ataupun sebagainya tanpa menggunakan sebuah kabel sebagai penghubungnya. Yang mana jaringan ini memanfaatkan kinerja udara yang dijadikan sebagai media untuk melakukan transmisi penghantar gelombang elektromagnetik. Bukan hanya digunakan untuk perangkat komputer saja, wireless juga kerap dipakai untuk melengkapi berbagai perangkat canggih yang saat ini sudah sering ditemui.

Pengertian *wireless* menurut para ahli :

1. Priyambodo mengatakan bahwa Wireless merupakan standar dari jaringan tanpa kabel atau yang dikenal dengan nama Wireless Networking dengan fungsi untuk menyempurnakan komponen-komponen pada jaringan internet agar terkoneksi atau agar terhubung dengan internet dengan mudah dan tanpa ribet.
2. Onno W. Purbo mengatakan bahwa Wireless merupakan standar yang sering dimanfaatkan untuk alat bantu komunikasi jaringan tanpa menggunakan sebuah kabel. Yang mana mendasari dari spesifikasi IEEE. Biasanya, jaringan tanpa kabel tersebut digunakan ataupun disharing secara bersama-sama dalam sebuah ruangan ataupun komunitas tertentu.

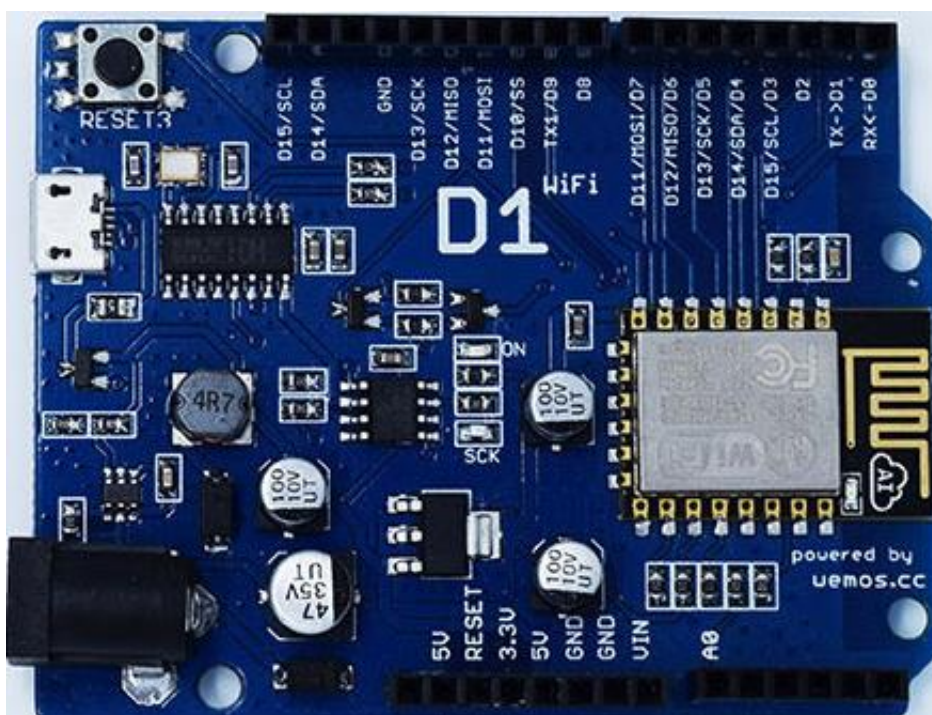
Berdasarkan kedua pendapat diatas, dapat disimpulkan bahwa *wireless* sendiri merupakan sebuah jaringan yang sudah banyak digunakan masyarakat dunia, untuk memudahkan terhubung dengan internet dan dapat digunakan ataupun di sharing secara bersama-sama dalam sebuah ruangan ataupun komunitas tertentu.

(sumber : <https://pengertiandefinisi.com/>, diakses tanggal 15 juni 2019)

## 2.3 **Microcontroller Wemos**

Microcontroller Wemos adalah sebuah Microcontroller pengembangan berbasis modul microcontroller ESP 8266. Microcontroller Wemos dibuat sebagai solusi dari mahalnya sebuah sistem wireless berbasis Microcontroller lainnya. Dengan menggunakan Microcontroller Wemos biaya

yang dikeluarkan untuk membangun sistem WiFi berbasis Microcontroller sangat murah, hanya sepersepuluhnya dari biaya yang dikeluarkan apabila membangun sistem WiFi dengan menggunakan Microcontroller Arduino Uno dan WiFi Shield Microcontroller Wemos yang berbeda pada Microcontroller ini yaitu kemampuannya untuk menyediakan fasilitas konektifitas WiFi dengan mudah serta memory yang digunakan sangat besar yaitu 4 MB. Berikut contoh dari gambar wemos yang dapat dilihat pada gambar 2.1.



**Gambar 2.1** Modul *Wemos*

(Sumber : <https://id.scribd.com/>, diakses tanggal 15 Juni 2019)

WeMos D1 merupakan module development board yang berbasis WiFi dari keluarga ESP8266 yang dimana dapat diprogram menggunakan software IDE Arduino seperti halnya dengan NodeMCU. Salah satu kelebihan dari WeMos D1 ini dibandingkan dengan module development board berbasis ESP8266 lainnya yaitu adanya module shield untuk pendukung hardware plug and play.

Module Shield Development yang dimaksud antara lain yang dapat dilihat pada tabel 2.1.

**Tabel 2.1. Module shield development**

OLED Shield	1-Button Shield
Motor Shield	Relay Shield
DHT Shield	ProtoBoard Shield
WS2812B RGB Shield	DC Power Shield
Battery LiPo Shield	DHT11 Shield
Buzzer Shield	Micro SC Card Shield

Meskipun bentuk board ini dirancang menyerupai arduino uno, namun dari sisi spesifikasi, sebenarnya jauh lebih unggul Wemos D1, salah satunya dikarenakan inti dari Wemos D1 adalah Esp8266EX yang memiliki prosesor 32 bit. (Bandingkan dengan Arduino UNO, yang berintikan AVR 8 bit). Sebagaimana board berbasis ESP8266, wemos D1 memiliki spesifikasi yang sama yaitu :

- A. A 32 bit RISC CPU running at 80MHz
- B. 64Kb of instruction RAM and 96Kb of data RAM
- C. 4MB flash memory! Yes that's correct, 4MB!
- D. Wi-Fi
- E. 16 GPIO pins
- F. I2C,SPI
- G. I2S
- H. 1 ADC

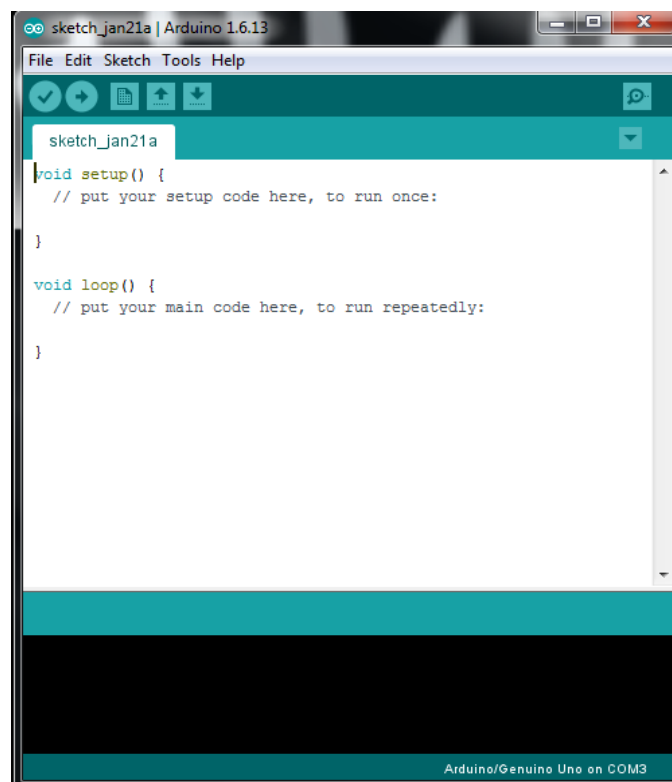
Berikut spesifikasi dari Modul *Wemos* yang dapat dilihat pada tabel 2.2.

**Tabel 2.2. Spesifikasi Modul *Wemos***

Spesifikasi	Modul <i>Wifi</i> ESP8266 Nodemcu
Mikrokontroler	Wemos D1
Ukuran Board	34,2 mm x 25,6 mm
Tegangan Input	3.3 ~ 5V

<b>GPIO</b>	11 Pin
<b>Kanal PWM</b>	10 Kanal
<b>10 bit ADC Pin</b>	1 Pin
<b>Flash Memori</b>	4 MB
<b>Clock Speed</b>	80 MHz
<b>IC</b>	CH340G
<b>Frekuensi</b>	2.4 Ghz – 22.5 GHz
<b>USB Port</b>	Micro USB

## 2.4 Software Arduino IDE



**Gambar 2.2** Tampilan Arduino IDE







Arduino *Development Environment* terdiri dari editor teks untuk menulis kode, sebuah area pesan, sebuah konsol, sebuah *toolbar* dengan tombol- tombol

untuk fungsi yang umum dan beberapa menu. *Arduino Development Environment* terhubung ke arduino board untuk meng-*upload* program dan juga untuk berkomunikasi dengan arduino board.

Arduino IDE dibuat dari bahasa pemrograman JAVA. Arduino IDE juga dilengkapi dengan *library C/C++* yang biasa disebut *Wiring* yang membuat operasi input dan output menjadi lebih mudah. Arduino IDE ini dikembangkan dari *software Processing* yang dirombak menjadi Arduino IDE khusus untuk pemrograman dengan Arduino.

Perangkat lunak yang ditulis menggunakan *Arduino Development Environment* disebut *sketch*. *Sketch* ditulis pada editor teks. *Sketch* disimpan dengan file berekstensi *.ino*. area pesan memberikan informasi dan pesan error ketika kita menyimpan atau membuka *sketch*. Konsol menampilkan output teks dari *Arduino Development Environment* dan juga menampilkan pesan error ketika kita mengkompile *sketch*. Pada sudut kanan bawah jendela *Arduino Development Environment* menunjukkan jenis board dan port serial yang sedang digunakan.

**Tabel 2.3** Fungsi *shortcut button* di Arduino IDE

No	Icon	Nama	Fungsi
1.		Verify	Untuk mengecek error kode program yang telah dibuat.
2.		Upload	Meng- <i>upload</i> program ke Arduino board.
3.		New	Untuk membuat <i>sketch</i> baru.
4.		Open	Membuka <i>sketch</i> program yang telah disimpan.
5.		Save	Menyimpan <i>sketch</i> program yang dibuat.
6.		Serial	Membuka layar serial

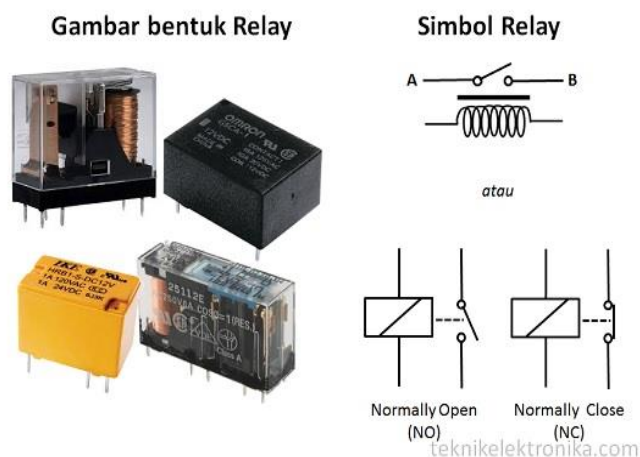
## 2.5 Relay

*Relay* adalah saklar mekanik yang dikendalikan atau dikontrol secara elektronika (elektro magnetik). Saklar pada *relay* akan terjadi perubahan posisi OFF ke ON pada saat diberikan energi elektro magnetik pada armatur relay tersebut. *Relay* pada dasarnya terdiri dari 2 bagian utama yaitu saklar mekanik dan sistem pembangkit elektromagnetik (induktor inti besi). Saklar atau kontaktor relay dikendalikan menggunakan tegangan listrik yang diberikan ke induktor pembangkit magnet untuk menarik armatur tuas saklar atau kontaktor *relay*.

*Relay* elektro mekanik memiliki kondisi saklar atau kontaktor dalam 3 posisi. Ketiga posisi saklar atau kontaktor relay ini akan berubah pada saat *relay* mendapat tegangan sumber pada elektromagnetnya. Ketiga posisi saklar *relay* tersebut adalah:

1. Posisi *Normally Open* (NO) , yaitu posisi saklar *relay* yang terhubung ke terminal NO (*Normally Open*). Kondisi ini akan terjadi pada saat *relay* mendapat tegangan sumber pada elektromagnetnya.
2. Posisi *Normally Colse* (NC), yaitu posisi saklar *relay* yang terhubung ke terminal NC (*Normally Close*). Kondisi ini terjadi pada saat *relay* tidak mendapat tegangan sumber pada elektromagnetnya.
3. Posisi *Change Over* (CO), yaitu kondisi perubahan armatur saklar *relay* yang berubah dari posisi NC ke NO atau sebaliknya dari NO ke NC.

Berikut contoh gambar relay yang dapat dilihat pada gambar 2.3.



**Gambar 2.3 Relay**

(Sumber : <http://www.teknikelektronika.com/>, diakses tanggal 15 juni 2019)

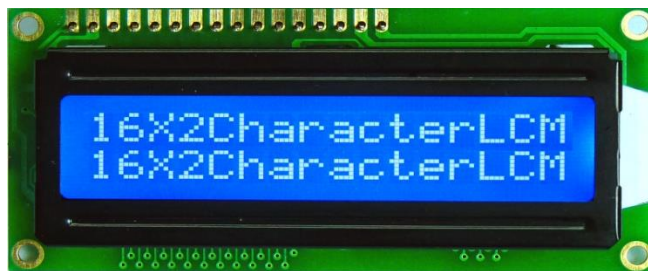
## 2.6 LCD

LCD (*Liquid Crystal Display*) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama, selain itu LCD juga dapat digunakan untuk menampilkan karakter atau gambar (boarduino.web.id).

Dalam modul LCD terdapat mikrokontroler yang berfungsi sebagai pengendali tampilan karakter LCD. Mikrokontroler pada suatu LCD dilengkapi dengan memori dan *register*, memori dan *register* yang digunakan adalah:

1. DDRAM (*Display Data Random Access Memory*) merupakan memori tempat karakter yang ditampilkan.
2. CGRAM (*Character Generator Random Access Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana bentuk dari karakter dapat berubah-ubah sesuai dengan keinginan.
3. CGROM (*Character Generator Read Only Memory*) merupakan memori untuk menggambarkan pola sebuah karakter dimana pola tersebut merupakan karakter dasar yang sudah ditentukan secara permanen oleh pabrikan pembuat LCD.
4. *Register* perintah, yaitu *register* yang berisi perintah-perintah dari mikrokontroler ke panel LCD pada saat proses penulisan data atau tempat status dari panel LCD dapat dibaca pada saat pembacaan data.
5. *Register* data, yaitu register menuliskan atau membaca data dari atau ke DDRAM. Penulisan data pada *register* akan menempatkan data tersebut ke DDRAM sesuai dengan alamat yang telah diatur.

Berikut contoh gambar dari LCD (*Liquid Crystal Display*), yang dapat dilihat pada Gambar 2.4.



**Gambar 2.4** *Liquid Crystal Display*

(Sumber: <http://www.boarduino.web.id/>, diakses tanggal 20 Maret 2019)



Pin, kaki atau jalur *input* dan kontrol dalam suatu LCD diantaranya adalah:

1. Pin data adalah jalur untuk memberikan data karakter yang ingin ditampilkan menggunakan LCD (*Liquid Cristal Display*) dapat dihubungkan dengan bus data dari rangkaian lain seperti mikrokontroler dengan lebar data 8 bit.
2. Pin RS (*Register Select*) berfungsi sebagai indikator atau yang menentukan jenis data yang masuk, apakah data atau perintah. Logika *low* menunjukkan yang masuk adalah perintah, sedangkan logika *high* menunjukkan data.
3. Pin R/W (*Read Write*) berfungsi sebagai instruksi pada modul jika *low* tulis data, sedangkan *high* baca data.
4. Pin E (*Enable*) digunakan untuk memegang data baik masuk atau keluar.
5. Pin VLCD berfungsi mengatur kecerahan tampilan (kontras) dimana pin ini dihubungkan dengan trimpot 5 Kohm, jika tidak digunakan dihubungkan ke ground, sedangkan tegangan catu daya ke LCD sebesar 5 Volt.

## 2.7 Solenoid Door Lock

Bagian ini berfungsi sebagai aktuator. Prinsip dari solenoid sendiri akan bekerja sebagai pengunci dan akan aktif ketika diberikan tegangan sebesar 12 V. Didalam solenoid terdapat kawat yang melingkar pada inti besi. Ketika arus listrik mengalir melalui kawat ini, maka terjadi medan magnet untuk menghasilkan energi yang akan menarik inti besi ke dalam ( Wibowo, 2014). Gambar dari solenoid door lock yang bisa dilihat pada gambar 2.5 :



**Gambar 2.5** Solenoid *Door Lock*

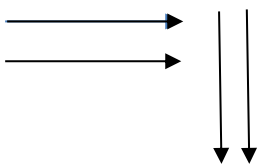
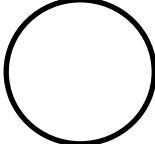
## 2.8 *Application Programming Interface (API)*


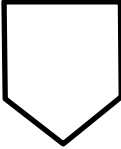

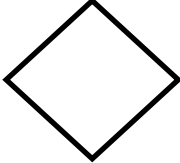
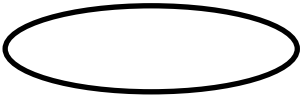


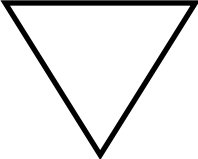
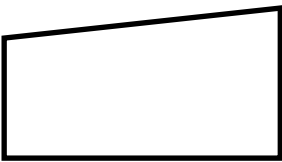
API adalah singkatan dari Application Programming Interface, dan memungkinkan developer untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. API terdiri dari berbagai elemen seperti function, protocols, dan tools lainnya yang memungkinkan developers untuk membuat aplikasi. Tujuan dari API adalah untuk mempercepat pembuatan suatu aplikasi. API juga disediakan oleh sebuah platform untuk dapat mengakses fitur dari platform tersebut. Contoh dari API yang lintas platform adalah API Twitter dan Facebook, yang memungkinkan untuk dapat mengakses data pengguna platform tersebut pada aplikasi. Contoh lain seperti API Bot Telegram dan LINE yang memungkinkan aplikasi untuk dapat mengirim dan membaca chat dari pengguna platform tersebut secara otomatis.


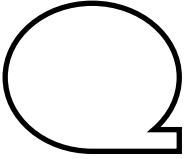



## 2.9 *Flowchart*

*Flowchart* atau diagram alir merupakan representasi grafik dari langkah-langkah yang harus diikuti dalam menyelesaikan suatu permasalahan yang terdiri dari sekumpulan simbol, dimana masing-masing simbol mempresentasikan suatu kegiatan tertentu. *Flowchart* diawali dengan menerima input, pemrosesan input, dan diakhiri dengan menampilkan output (Agus, 2014).

**Tabel 2.4** Simbol-simbol dan keterangan *Flowchart*

No	Simbol	Keterangan
1.		Simbol arus / <i>flow</i> , yaitu menyatakan jalannya arus suatu program.
2.		Simbol <i>connector</i> , berfungsi menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.

3.		Simbol <i>manual</i> , yaitu menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.
4.		Simbol <i>offline connector</i> , menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
5.		Simbol <i>process</i> , menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
6.		Simbol <i>decision</i> , yaitu menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban ya/tidak.
7.		Simbol <i>terminal</i> , yaitu menyatakan permulaan atau akhir suatu program.
8.		Simbol <i>predefined process</i> , yaitu menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
9.		Simbol <i>keying operation</i> , menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.
10.		Simbol <i>offline-storage</i> , menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu.
11.		Simbol <i>manual input</i> , memasukkan data secara manual dengan menggunakan online keyboard.

12.		Simbol <i>input/output</i> , menyatakan proses masukan atau keluaran tanpa tergantung jenis peralatannya.
13.		Simbol <i>magnetic tape</i> , menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis.
14.		Simbol <i>disk storage</i> , menyatakan input berasal dari disk atau output disimpan ke disk.
15.		Simbol <i>document</i> , mencetak keluaran dalam bentuk dokumen.
16.		Simbol <i>display</i> , mencetak keluaran dalam layar monitor.