

BAB II TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Relevansi antara penelitian terdahulu dengan penelitian penulis. Berikut pada table 2.1 merupakan Tabel Penelitian Terdahulu.

Tabel 2. 1 Penelitian Terdahulu

No.	Judul Referensi	Penulis	Tahun	Metode Penelitian	Hasil Penelitian
1.	<i>Smart Home</i> Monitoring Pintu Rumah dengan Identifikasi Wajah menerapkan <i>Camera</i> ESP32 berbasis IoT	Budi Yanto, Basorudin, Syaiful Anwar, Adyanata Lubis, Karmi	2022	Metode yang digunakan adalah metode pengujian <i>blackbox</i> .	Penelitian ini berhasil merancang alat proteksi wajah pada pintu menggunakan <i>camera</i> ESP32Cam. Alat ini dapat membuka pintu melalui sensor wajah.
2.	Rancang Bangun Dual Keamanan Sistem Pintu Rumah menggunakan Pengenalan Wajah dan Sidik Jari berbasis IoT (<i>Internet of Things</i>)	Ahmad Haris Bachtiar, Pressa Perdana Surya, Rini Puji Astutik	2022	Metode yang digunakan adalah metode prototipe	Sistem Keamanan Pintu Rumah dengan metode pengenalan wajah dan sidik jari.
3.	Perancangan Sistem Pintu Otomatis menggunakan ESP32CAM	Suradi, Ahmad Martani, Irma Arfiani, Sarli	2022	Metode yang digunakan adalah metode <i>Research and Development</i>	Rancangan <i>Prototype</i> system keamanan pintu dengan ESP32CAM

No.	Judul Referensi	Penulis	Tahun	Metode Penelitian	Hasil Penelitian
4.	Rancang Bangun Pintu Otomatis dengan Menggunakan RFID	Triuli Novianti	2019	Metode yang digunakan pada penelitian ini adalah metode Riset dan pengembangan.	<i>Prototype</i> Pintu otomatis menggunakan RFID.
5.	Rancang Bangun <i>Magnetic Door Lock</i> menggunakan <i>Keypad</i> dan Selenoid berbasis Mikrokontroler Arduino Uno	Helmi Guntoro, Yoyo Somantri, Erik Haritman	2013	Penelitian ini menggunakan metode eksperimen (uji coba).	Hasil dari penelitian ini adalah Kunci Pintu dengan akses masuk Pin

2.2 *Internet of Things*

Internet of things merupakan teknologi baru dalam internet akses yang dapat mengenali objek perilaku intelijen terkait dengan pengambilan suatu keputusan dan dapat berkomunikasi dengan dirinya sendiri. IoT dapat menghubungkan berbagai objek tidak hidup melalui koneksi *internet* dan dapat menghubungkan mereka untuk berbagi informasi dan dapat melakukan proses otomatis (Ahdan & Susanto, 2021).

IoT (*Internet of things*) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas internet yang tersambung secara terus menerus. Pada dasarnya IoT (*Internet of things*) mengacu pada benda yang dapat diidentifikasi secara unik sebagai *representative* virtual dalam struktur berbasis internet. Cara kerja IoT (*Internet of things*) adalah interaksi antara sesama mesin yang terhubung secara otomatis tanpa campur tangan *user* dan dalam jarak berapapun (Skad & Nandika, 2020).

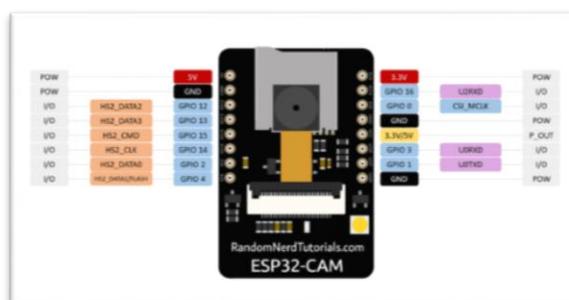
Internet of Things adalah penggabungan teknologi seperti sensor dan perangkat lunak ke dalam satu atau lebih objek untuk tujuan berkomunikasi,

mengontrol, menghubungkan, dan bertukar data melalui perangkat lain selama mereka terhubung ke Internet. IoT erat kaitannya dengan istilah mesin mesin atau M2M. Semua alat dengan kemampuan komunikasi M2M sering disebut sebagai perangkat pintar.

2.3 Mikrokontroler ESP32-CAM

ESP32-CAM merupakan salah satu mikrokontroler yang memiliki fasilitas tambahan berupa *bluetooth*, *wifi*, kamera, bahkan sampai ke slot microSD. ESP32-CAM ini biasanya digunakan untuk project IoT (*Internet of Things*) yang membutuhkan fitur kamera. Modul ESP32CAM memiliki lebih sedikit pin I/O dibandingkan modul ESP32 produk sebelumnya, yaitu ESP32 Wroom. Hal ini dikarenakan sudah banyak pin yang digunakan secara internal untuk fungsi kamera dan fungsi slot kartu *microSD*. Selain itu, modul ESP32CAM juga tidak memiliki *port* USB khusus (mengirim program dari *port* USB komputer). Jadi untuk memprogram modul ini Anda harus menggunakan USB TTL atau kita dapat menambahkan modul tambahan berupa *downloader* khusus untuk ESP32- CAM (Setiawan.P, 2022).

ESP32-CAM yang dilengkapi dengan *camera* OV2640 digunakan untuk mengambil data dari sensor, mengambil gambar sebagai dokumentasi visual keberadaan manusia, mendeteksi wajah (*face detection*) dan pengenalan wajah (*face recognition*) ESP32- CAM terhubung dengan *Wi-Fi* untuk koneksi internet. Dari beberapa IoT sederhana, hingga lainnya yang lebih canggih untuk pemantauan dan pengenalan gambar menggunakan AI, dan bahkan sebagai sistem pengawasan untuk memeriksa apa yang terjadi di suatu tempat dari jarak jauh di mana pun berada Berikut pada gambar 2.4 merupakan gambar ESP32 – CAM.



Gambar 2. 1 Esp-32 CAM

2.4 *Arduino Integrated Development Environment (IDE)*

Integrited Development Enviroenment (IDE) merupakan program yang digunakan untuk membuat program pada Arduino Uno Program yang ditulis dengan menggunakan *Software Arduino (IDE)* disebut sebagai *sketch*. *Sketch* ditulis dalam suatu editor teks dan disimpan dalam file dengan ekstensi *ino*. Pada *Software Arduino IDE*, terdapat semacam *message box* berwarna hitam yang berfungsi menampilkan status, seperti pesan error, *compile*, dan *upload* program. Di bagian bawah paling kanan *software Arduino IDE*, menunjukkan *board* yang terkonfigurasi beserta *COM Ports* yang digunakan.

1. *Verify/Compile* berfungsi untuk mengecek apakah *sketch* yang dibuat ada kekeliruan dari segi sintaks atau tidak. Jika tidak ada kesalahan, maka sintaks yang dibuat akan *dcompile* ke dalam bahasa mesin.
2. *Upload* berfungsi mengirimkan program yang sudah dikompilasi ke *Arduino Board* (Shofiyullah, 2020).



Gambar 2. 2 Tampilan Arduino IDE

(Sumber : <https://www.kmtech.id/post/mengenal-perangkat-lunak-arduino-ide>)

Menurut Calvin (2020) Arduino Uno merupakan lingkungan terintegrasi yang digunakan untuk melakukan pengembangan. Pada tampilan awal arduino IDE terdapat tombol *verify* dapat mengkompilasi program yang ada di editor, Tombol *New* memiliki fungsi membuat program baru dengan mengosongkan isi dari jendela

editor. IDE memberikan kesempatan untuk menyimpan semua perubahan yang sebelumnya belum di *save*. Ketika mengklik tombol *upload* Arduino IDE mengkompilasi program dan *upload* ke papan arduino uno yang telah dipilih di IDE menu *Tools* lalu ke *serial port*.

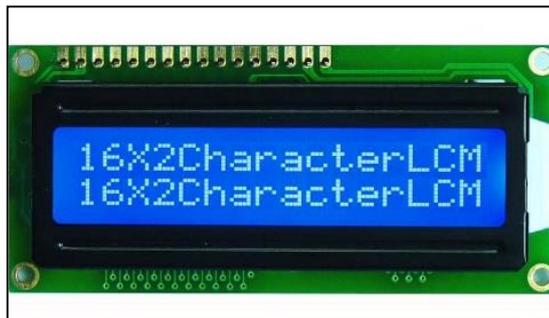
2.5 *Liquid Crystal Display*

LCD (*Liquid Cristal Display*) adalah salah satu jenis *display* elektronik yang dibuat dengan teknologi CMOS *logic* yang bekerja dengan tidak menghasilkan cahaya tetapi memantulkan cahaya yang ada di sekelilingnya terhadap *front-lit* atau mentransmisikan cahaya dari *back-lit*. LCD berfungsi sebagai penampil data baik dalam bentuk karakter, huruf, angka ataupun grafik. LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan indium oksida dalam bentuk tampilan *seven-segment* dan lapisan elektroda pada kaca belakang. Ketika elektroda diaktifkan dengan medan listrik (tegangan), molekul organik yang panjang dan silindris menyesuaikan diri dengan elektroda dari segmen. Lapisan *sandwich* memiliki *polarizer* cahaya vertikal depan dan *polarizer* cahaya horizontal belakang yang diikuti dengan lapisan *reflektor*. Cahaya yang dipantulkan tidak dapat melewati molekul-molekul yang telah menyesuaikan diri dan segmen yang diaktifkan terlihat menjadi gelap dan membentuk karakter data yang ingin ditampilkan (Natsir, 2019).

Menurut Royhan (2018) Layar LCD merupakan suatu media penampilan data yang sangat efektif dan efisien dalam penggunaannya. Untuk menampilkan sebuah karakter pada layar LCD diperlukan beberapa rangkaian tambahan. Untuk lebih memudahkan para pengguna, maka beberapa perusahaan elektronik menciptakan modul LCD Tampilan LCD untuk menampilkan angka atau teks. dua jenis LCD *Display*. LCD yang digunakan untuk tampilan pengaturan menggunakan LCD 16x2 LCD dibagi menjadi dua bagian yaitu bagian depan panel LCD yang terdiri dari banyak dot atau titik LCD dan mikrokontroler yang menempel pada bagian belakang panel LCD yang berfungsi untuk mengatur titik-titik LCD sehingga dapat menampilkan huruf, angka, dan simbol khusus yang dapat terbaca.

2.5.1 Karakteristik LCD 16x2

1. 16 karakteristik x 2 baris
2. 5x7 titik matriks karakter + kursor
3. HD44780 *equivalent* LCD controller/diver built-in
4. 4 bit atau 8 bit MPU *interface* Tipe standar
5. Bekerja hampir semua mikrokontroler (Royhan, 2018).



Gambar 2. 3 Bentuk Fisik LCD 16 x 2

(Sumber : <https://www.nyebarilmu.com/cara-mengakses-modul-display-lcd-16x2/>)

2.5.2 Spesifikasi LCD 16x2

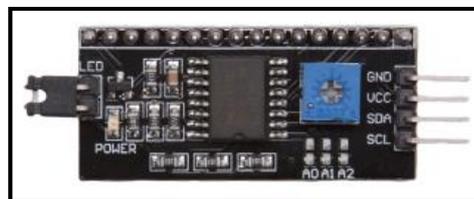
Tabel 2. 2 Spesifikasi LCD 16x2

Pin	Simbol	Fungsi
1	Vss	Ground
2	Vdd	+3V Atau +5V
3	Vo	Pengatur Kontras
4	Rs	H/L Register Select Signal
5	R/W	Read/Write Signal
6	EN	Enable Signal
7	Data	I/O Pins
8	Anoda	Tegangan Positif
9	Katoda	Tegangan Negatif

2.6 *Inter Integrated Circuit (I2C)*

Menurut M.Natsir,dkk (2019) I2C/TWI LCD merupakan modul yang dipakai untuk mengurangi penggunaan kaki di LCD. Modul ini memiliki 4 pin yang akan dihubungkan ke Arduino. Arduino uno sudah mendukung komunikasi I2C dengan module I2C lcd, maka dapat mengontrol LCD Karakter 16x2 dan 20x4 hanya menggunakan 2 Pin yaitu Analog *Input* Pin 4 (SDA) dan Analog *Input* Pin 5 (SCL).

Inter Integrated Circuit atau sering disebut I2C adalah standar komunikasi serial dua arah menggunakan dua saluran yang didesain khusus untuk mengirim maupun menerima data. Sistem I2C terdiri dari saluran SCL (*Serial Clock*) dan SDA (*Serial Data*) yang membawa informasi data antara I2C dengan pengontrolnya. Piranti yang dihubungkan dengan sistem I2C Bus dapat dioperasikan sebagai *Master* dan *Slave*. *Master* adalah piranti yang memulai *transfer* data pada I2C Bus dengan membentuk sinyal *start*, mengakhiri *transfer* data dengan membentuk sinyal *stop*, dan membangkitkan sinyal *clock*. *Slave* adalah piranti yang dialamati *master* (Yohanes, 2018).

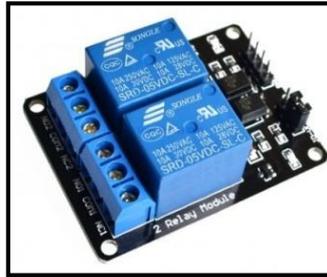


Gambar 2. 4 *Inter Integrated Circuit (I2C)*

(Sumber: <https://www.nyebarilmu.com/cara-mengakses-modul-display-lcd-16x2/>)

2.7 **Relay**

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni *Electromagnet (Coil)* dan Mekanikal (seperangkat Kontak Saklar/*Switch*). Relay menggunakan Prinsip Elektromagnetik untuk menggerakkan Kontak Saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi.



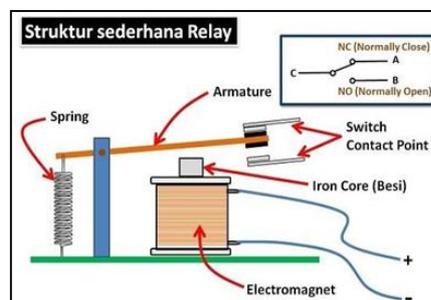
Gambar 2. 5 Relay

(Sumber : <https://images.app.goo.gl/r6o7NrhVfPy4DP3u7>)

Pada dasarnya, Relay terdiri dari 4 komponen dasar yaitu :

1. *Electromagnet (Coil)*
2. *Armature*
3. *Switch Contact Point (Saklar)*
4. *Spring*

Berikut ini merupakan gambar dari bagian-bagian Relay :



Gambar 2. 6 Bagian-bagian Relay

(Sumber : <https://www.aldyrazor.com/2020/05/modul-relay-arduino.html>)

Kontak Poin (*Contact Point*) Relay terdiri dari 2 jenis yaitu:

- *Normally Close (NC)* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *CLOSE* (tertutup).
- *Normally Open (NO)* yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *OPEN* (terbuka).

Karena Relay merupakan salah satu jenis dari Saklar, maka istilah *Pole* dan *Throw* yang dipakai dalam Saklar juga berlaku pada Relay. Berikut ini adalah penjelasan singkat mengenai Istilah *Pole* and *Throw* :

- *Pole* : Banyaknya Kontak (*Contact*) yang dimiliki oleh sebuah relay
- *Throw* : Banyaknya kondisi yang dimiliki oleh sebuah Kontak (*Contact*)

Berdasarkan penggolongan jumlah *Pole* dan *Throw*-nya sebuah relay, maka relay dapat digolongkan menjadi :

- *Single Pole Single Throw (SPST)* : Relay golongan ini memiliki 4 Terminal, 2 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil*.
- *Single Pole Double Throw (SPDT)* : Relay golongan ini memiliki 5 Terminal, 3 Terminal untuk Saklar dan 2 Terminalnya lagi untuk *Coil*.
- *Double Pole Single Throw (DPST)* : Relay golongan ini memiliki 6 Terminal, diantaranya 4 Terminal yang terdiri dari 2 Pasang Terminal Saklar sedangkan 2 Terminal lainnya untuk *Coil*. Relay DPST dapat dijadikan 2 Saklar yang dikendalikan oleh 1 *Coil*.
- *Double Pole Double Throw (DPDT)* : Relay golongan ini memiliki Terminal sebanyak 8 Terminal, diantaranya 6 Terminal yang merupakan 2 pasang Relay SPDT yang dikendalikan oleh 1 (*single*) *Coil*. Sedangkan 2 Terminal lainnya untuk *Coil*.

Selain Golongan Relay diatas, terdapat juga Relay-relay yang *Pole* dan *Throw*-nya melebihi dari 2 (dua). Misalnya 3PDT (*Triple Pole Double Throw*) ataupun 4PDT (*Four Pole Double Throw*) dan lain sebagainya. (Saleh, 2017) .

Seperti yang telah dijelaskan tadi bahwa relay memiliki fungsi sebagai saklar elektrik, namun jika di aplikasikan ke dalam rangkaian elektronika, relay memiliki beberapa fungsi yang cukup unik. Berikut beberapa fungsi saat di aplikasikan ke dalam sebuah rangkaian elektronika.

1. Relay digunakan untuk menjalankan Fungsi Logika (*Logic Function*)
2. Relay digunakan untuk memberikan Fungsi penundaan waktu (*Time Delay Function*)
3. Relay digunakan untuk mengendalikan Sirkuit Tegangan tinggi dengan bantuan dari Signal Tegangan rendah (Saleh, 2017).

2.8 Solenoid Door Lock

Menurut (Yohanes dkk, 2018) Solenoid Door Lock adalah salah satu solenoid yang difungsikan khusus sebagai solenoid untuk pengunci pintu secara elektronik. Solenoid ini mempunyai dua sistem kerja, yaitu Normaly Close (NC) dan Normaly Open (NO). Perbedaanya adalah jika cara kerja solenoid NC apabila diberi tegangan, maka solenoid akan memanjang (tertutup). Dan untuk cara kerja dari Solenoid NO adalah kebalikannya dari Solenoid NC. Biasanya kebanyakan Solenoid Door Lock membutuhkan input atau tegangan kerja 12V DC, tetapi ada juga Solenoid Door Lock yang hanya membutuhkan input tegangan 5V DC dan sehingga dapat langsung bekerja dengan tegangan output dari pin IC digital. Namun jika menggunakan Solenoid Door Lock yang 12V DC, Pada kondisi normal solenoid dalam posisi tuas memanjang / terkunci. Jika diberi tegangan tuas akan memendek/terbuka. Solenoid ini bisa digabungkan dengan sistem pengunci elektrik berbasis RFID dan password. Cocok dipakai untuk pengunci pintu ataupun locker/lemari. Membutuhkan power supply 12V dan sebuah relay untuk mengaktifkannya.



Gambar 2. 7 Solenoid Door lock

2.9 Push Button Switch

Push button switch (saklar tombol tekan) adalah perangkat / saklar sederhana yang berfungsi untuk menghubungkan atau memutuskan aliran arus listrik dengan sistem kerja tekan *unlock* (tidak mengunci). Sistem kerja *unlock* disini berarti saklar akan bekerja sebagai device penghubung atau pemutus aliran arus listrik saat tombol ditekan, dan saat tombol tidak ditekan (dilepas), maka saklar akan kembali pada kondisi normal.



Gambar 2. 8 *Push Button Swicth*

Gambar 1 *Push button switch* Sebagai *device* penghubung atau pemutus, *push button switch* hanya memiliki 2 kondisi, yaitu *On* dan *Off* (1 dan 0). Istilah *On* dan *Off* ini menjadi sangat penting karena semua perangkat listrik yang memerlukan sumber energi listrik pasti membutuhkan kondisi *On* dan *Off*.

Karena sistem kerjanya yang *unlock* dan langsung berhubungan dengan operator, *push button switch* menjadi *device* paling utama yang biasa digunakan untuk memulai dan mengakhiri kerja mesin di industri. Secanggih apapun sebuah mesin bisa dipastikan sistem kerjanya tidak terlepas dari keberadaan sebuah saklar seperti *push button switch* atau perangkat lain yang sejenis yang bekerja mengatur pengkondisian *On* dan *Off* (Riski, M. D. 2019).

2.10 Blynk

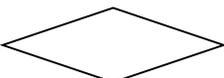
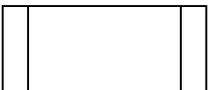
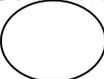
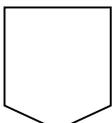
Blynk adalah IoT *Cloud* platform untuk aplikasi iOS dan Android yang berguna untuk mengontrol Arduino, Raspberry Pi, dan *board-board* sejenisnya melalui Internet. Blynk adalah *dashboard* digital di mana anda dapat membangun sebuah antarmuka grafis untuk alat yang telah dibuat hanya dengan menarik dan menjatuhkan sebuah *widget*. Blynk sangat mudah dan sederhana untuk mengatur semuanya dan hanya dalam waktu kurang dari 5 menit. Blynk tidak terikat dengan beberapa mikrokontroler tertentu atau *shield* tertentu. (Gunawan, D , 2018).

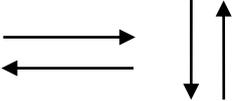
Terdapat 3 komponen utama di platform Blynk yaitu Blynk App yang digunakan untuk membuat *interface* dengan *widget* yang disediakan, Blynk Server yang bertanggung jawab tentang semua komunikasi antara *smartphone* dan *hardware*, dan Blynk *Libraries* yang digunakan untuk komunikasi antara server dengan proses *INPUT* dan *OUTPUT*.

2.11 Flowchart

Flowchart adalah representasi secara simbolik dari suatu algoritma atau prosedur untuk menyelesaikan suatu masalah, dengan menggunakan *flowchart* akan memudahkan pengguna melakukan pengecekan bagian-bagian yang terlupakan dalam analisis masalah, disamping itu *flowchart* juga berguna sebagai fasilitas untuk berkomunikasi antara pemrogram yang bekerja dalam tim suatu proyek. *Flowchart* membantu memahami urutan-urutan logika yang rumit dan panjang. *Flowchart* membantu mengkomunikasikan jalannya program ke orang lain (bukan pemrogram) akan lebih mudah (Santoso, 2017).

Tabel 2. 3 Simbol-Simbol Flowchart

Simbol	Keterangan
Terminal 	Digunakan untuk menunjukkan awal dan akhir dari program
Persiapan 	Digunakan untuk memberikan nilai pada awal suatu variabel atau <i>counter</i>
Proses 	Digunakan untuk mengolah aritmatika dan pemindahan data
Keputusan 	Digunakan untuk mewakili operasi perbandingan logika
Proses 	Digunakan untuk proses yang detailnya dijelaskan terpisah, misalnya dalam bentuk <i>subroutine</i>
Connector 	Digunakan untuk menunjukan hubungan arus proses yang terputus masih dalam halaman yang sama
Penghubung 	Digunakan untuk menunjukan hubungan arus dari suatu proses yang terputus dalam halaman yang berbeda

Simbol	Keterangan
<p data-bbox="448 412 515 439">Arus</p> 	<p data-bbox="660 468 1121 501">Penghubung antar prosedur / proses</p>
<p data-bbox="413 600 550 627"><i>Document</i></p> 	<p data-bbox="660 638 1348 707">Simbol yang menyatakan <i>input</i> berasal dari dokumen dalam bentuk kertas atau <i>output</i> di cetak dikertas</p>
<p data-bbox="395 786 568 813"><i>Input-Output</i></p> 	<p data-bbox="660 817 1299 887">Simbol yang menyatakan proses <i>input</i> dan <i>output</i> tanpa tergantung dengan jenis peralatannya</p>
<p data-bbox="395 960 568 987"><i>Disk Storage</i></p> 	<p data-bbox="660 996 1353 1066">Simbol untuk menyatakan <i>input</i> berasal dari <i>disk</i> atau <i>output</i> disimpan ke <i>disk</i>.</p>