

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Penelitian mengenai sistem *Locker* berbasis teknologi telah banyak dikembangkan untuk meningkatkan keamanan dan efisiensi penyimpanan barang. Menurut Ananda et al. (2023), *Locker* konvensional yang masih menggunakan kunci fisik memiliki beberapa kelemahan, seperti risiko kehilangan kunci, potensi duplikasi, serta proses pengelolaan yang tidak efisien. Untuk mengatasi permasalahan tersebut, dikembangkan sistem *Locker* berbasis Internet of Things (IoT) dengan autentikasi *QR Code* yang memungkinkan proses akses dan monitoring dapat dilakukan melalui aplikasi mobile. Sistem tersebut memanfaatkan mikrokontroler sebagai pengendali utama dan kunci elektronik sebagai pengganti kunci fisik sehingga penggunaan *Locker* menjadi lebih aman dan praktis.

Penelitian lain oleh Suryani, Lindawati, dan Fadhli (2024) menunjukkan bahwa penggunaan *QR Code* pada sistem akses seperti smart parking dapat meningkatkan keaslian pengguna serta mengurangi kecurangan melalui proses verifikasi digital secara real-time. Sementara itu, Pratama (2022) menjelaskan bahwa *RFID* merupakan teknologi identifikasi berbasis kartu yang memberikan kemudahan akses namun memiliki kelemahan seperti risiko kartu hilang atau rusak. Oleh karena itu, menggabungkan metode autentikasi *QR Code* dan *RFID* dinilai dapat meningkatkan fleksibilitas sekaligus keamanan dalam sistem akses *Locker*. Berdasarkan penelitian-

penelitian tersebut, sistem yang dikembangkan pada penelitian ini menggabungkan IoT, *QR Code*, *RFID*, dan Sensor *Ultrasonik* untuk menghasilkan *Locker* yang aman, efisien, dan dapat dipantau secara real-time.

2.1.1 Locker

Locker merupakan fasilitas penyimpanan barang yang umum digunakan di perpustakaan, sekolah, pusat olahraga, dan fasilitas umum lainnya sebagai tempat penyimpanan sementara untuk menjaga keamanan barang pengguna. Pada awalnya, sistem *Locker* masih bergantung pada kunci fisik, namun metode ini memiliki beberapa kelemahan, seperti risiko kehilangan kunci, kemungkinan duplikasi, serta kesulitan dalam pengelolaan secara manual. Seiring dengan perkembangan teknologi, sistem *Locker* mulai dikembangkan menjadi sistem otomatis dengan metode autentikasi digital seperti *QR Code* dan *RFID*, yang mampu meningkatkan keamanan, efisiensi, serta kemudahan akses bagi pengguna (Ananda et al., 2023; Suryani et al., 2024).

2.1.2 Smart

Istilah *Smart* dalam konteks teknologi merujuk pada sistem atau perangkat yang memiliki kemampuan untuk mengumpulkan data, memproses informasi, serta mengambil keputusan secara otomatis berdasarkan kondisi tertentu melalui integrasi perangkat keras, perangkat lunak, dan konektivitas jaringan. Konsep *Smart* umumnya dikaitkan dengan penerapan teknologi berbasis Internet of Things (IoT), di mana suatu sistem mampu berkomunikasi, melakukan pemantauan secara *real-time*, serta merespons perubahan lingkungan secara adaptif dan efisien. Dengan adanya

kecerdasan terintegrasi ini, sistem *Smart* tidak hanya berfungsi secara pasif, tetapi juga mampu melakukan otomatisasi dan optimasi proses operasional untuk meningkatkan efisiensi serta keamanan dalam berbagai bidang aplikasi (Rahaman, 2024).

Berdasarkan konsep tersebut, sistem *Smart Locker* dapat dikategorikan sebagai sistem *Smart* karena telah mengintegrasikan perangkat keras seperti mikrokontroler, sensor, dan modul komunikasi dengan perangkat lunak berbasis web serta koneksi internet untuk melakukan pengelolaan akses secara otomatis. *Smart Locker* tidak hanya berfungsi sebagai tempat penyimpanan biasa, tetapi mampu melakukan autentikasi pengguna melalui *RFID* atau *QR Code*, mencatat riwayat akses secara otomatis ke dalam database, serta memungkinkan proses monitoring dan manajemen data secara *real-time* oleh admin. Kemampuan sistem dalam mengumpulkan data, memproses informasi akses, serta mengontrol kunci loker secara otomatis menunjukkan bahwa *Smart Locker* memiliki karakteristik sistem cerdas (*intelligent system*) yang adaptif, efisien, dan berbasis data sesuai dengan konsep teknologi *Smart*.

2.1.3 QR Code

QR Code (Quick Response Code) adalah kode dua dimensi yang mampu menyimpan informasi dalam jumlah yang lebih besar dibandingkan barcode satu dimensi, serta dapat dipindai dengan cepat menggunakan kamera smartphone. *QR Code* memiliki sifat unik, sulit dipalsukan, dan praktis, sehingga cocok digunakan sebagai metode autentikasi pada sistem akses *Locker*. Dalam penerapannya, pengguna

cukup melakukan pemindaian *QR Code* untuk membuka atau menutup *Locker* tanpa memerlukan kunci fisik atau PIN manual. Selain itu, *QR Code* dapat dihasilkan secara dinamis sehingga meningkatkan tingkat keamanan akses sistem (Agus & Candra, 2023; Ananda et al., 2023).



Gambar 2. 1 *QR Code* ([Bi.go.id](https://bi.go.id), 2025)

2.1.4 *RFID* (Radio Frequency Identification)

RFID (Radio Frequency Identification) adalah teknologi identifikasi berbasis gelombang radio yang memungkinkan pembacaan informasi pada kartu atau tag tanpa memerlukan kontak langsung dengan pembaca (reader). *RFID* digunakan sebagai opsi autentikasi alternatif selain *QR Code* karena proses aksesnya yang cepat dan mudah, yaitu cukup dengan menempelkan kartu *RFID* pada pembaca. Penggunaan *RFID* memberikan fleksibilitas bagi pengguna yang lebih nyaman menggunakan kartu fisik sebagai kunci akses (Pratama, 2022; Tawakal & Ramdhani, 2021).

2.1.5 Internet of Things

IoT atau dikenal sebagai *Internet of Things* merupakan sebuah gagasan yang bertujuan untuk memperluas layanan konektivitas internet yang selalu aktif. Konsep ini memungkinkan mesin, perangkat, dan objek fisik seperti sensor serta aktuator untuk terhubung ke jaringan internet guna memperoleh informasi, memantau kinerja, saling bekerja sama, serta merespons informasi yang diperoleh secara mandiri tanpa intervensi manusia (Rahaman et al., 2024).

2.1.5.1 Konsep Dari Internet of Things

Konsep *Internet of Things* ini sangat sederhana karena berfokus pada tiga komponen utama arsitekturnya, yaitu:

1. Barang fisik yang memiliki modul *Internet of Things*.
2. Perangkat yang dapat terhubung ke internet seperti modem atau router nirkabel yang cepat seperti WiFi.
3. Pusat Data Cloud tempat aplikasi dan database disimpan.

2.1.5.2 Manfaat Teknologi IoT Dalam Sistem *Smart Locker*

Penerapan teknologi Internet of Things (IoT) pada sistem *Smart Locker* memberikan berbagai manfaat dalam hal pengelolaan, keamanan, dan kemudahan penggunaan. IoT memungkinkan setiap komponen seperti mikrokontroler, sensor, dan aplikasi mobile untuk saling terhubung dan bertukar data secara real-time (Ananda et

al., 2023). Dengan adanya konektivitas ini, status *Locker* seperti kondisi *kosong* atau *terisi* dapat dipantau langsung melalui aplikasi tanpa memerlukan pengecekan manual. Hal ini meningkatkan efisiensi operasional dan mengurangi beban petugas dalam mengelola fasilitas penyimpanan barang.

Selain itu, IoT mendukung proses autentikasi digital menggunakan *QR Code* dan *RFID* yang terhubung dengan basis data pusat. Ketika pengguna memindai *QR Code* atau menempelkan kartu *RFID*, sistem dapat melakukan verifikasi secara otomatis sehingga proses akses menjadi lebih cepat dan aman (Suryani, Lindawati, & Fadhli, 2024). Metode ini lebih efektif dibandingkan kunci fisik yang berisiko hilang dan mudah diduplikasi. *RFID* juga memberikan fleksibilitas akses, meskipun masih memiliki kelemahan seperti risiko kerusakan kartu (Pratama, 2022). Selain itu, IoT memungkinkan penyimpanan riwayat penggunaan *Locker* sehingga pengelola dapat melakukan pemantauan dan audit keamanan barang secara sistematis. Dengan demikian, penerapan IoT menjadikan sistem *Smart Locker* lebih aman, efisien, modern, dan sesuai dengan kebutuhan lingkungan kampus.

2.1.6 Sensor

Sensor adalah perangkat yang digunakan untuk mendeteksi perubahan kondisi fisik, kimia, atau lingkungan tertentu dan mengubahnya menjadi sinyal listrik yang dapat diproses oleh sistem elektronik atau mikrokontroler. Dalam sistem berbasis Internet of Things (IoT), sensor berperan sebagai komponen input yang bertugas mengumpulkan data dari lingkungan sekitar, seperti suhu, kelembapan, jarak, cahaya,

maupun gerakan, yang kemudian dikirimkan untuk dianalisis dan diproses lebih lanjut oleh sistem. Keberadaan sensor memungkinkan sistem bekerja secara otomatis, responsif, dan berbasis data, sehingga meningkatkan efisiensi dan akurasi pengambilan keputusan dalam berbagai aplikasi teknologi modern (Rahaman, 2024).

2.1.7 Mikrokontroler, Sensor, Akuator Terkait Sistem *Smart Locker*

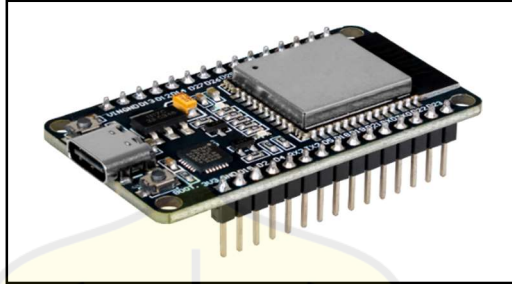
Internet of Things (IoT) merupakan konsep teknologi yang menghubungkan berbagai perangkat fisik seperti *mikrokontroler*, sensor, dan *aktuator* melalui jaringan internet sehingga perangkat-perangkat tersebut dapat saling bertukar data, memantau kondisi lingkungan, serta melakukan aksi secara otomatis tanpa membutuhkan interaksi manusia secara langsung (Rahaman et al., 2024).

2.1.7.1 Mikrokontroler *ESP32*

ESP32 adalah mikrokontroler *System-on-Chip* yang memiliki fitur Wi-Fi dan Bluetooth sehingga mendukung penerapan sistem berbasis Internet of Things (IoT). Pada penelitian ini, *ESP32* berfungsi sebagai pusat pengendali yang memproses input dari sistem autentikasi seperti pembacaan *QR Code* dan *RFID* untuk menentukan akses buka dan tutup *Locker* (Suryani et al., 2024).

Selain itu, *ESP32* mengelola komunikasi data secara real-time melalui jaringan Wi-Fi sehingga status *Locker* dapat dipantau melalui aplikasi mobile. Mikrokontroler ini juga mengontrol Sensor *Ultrasonik*, solenoid lock, buzzer, dan LCD Display,

sehingga keseluruhan proses penyimpanan dan pengambilan barang dapat berjalan otomatis dan efisien.



Gambar 2. 2 ESP32 (<https://joy-it.net> , 2025)

2.1.7.2 *ESP32-CAM*

ESP32-CAM adalah varian *ESP32* yang dilengkapi modul kamera dan konektivitas Wi-Fi, sehingga dapat digunakan untuk pemindaian *QR Code* secara langsung. Modul ini mendukung autentikasi berbasis gambar dengan proses verifikasi yang cepat melalui jaringan IoT (Tawakal & Ramdhani, 2021).

Dalam proyek *Smart Locker* ini, *ESP32-CAM* berfungsi sebagai pemindai *QR Code* untuk mengidentifikasi pengguna saat membuka *Locker*. Hasil pemindaian kemudian dikirim ke *ESP32* utama untuk menentukan apakah akses diperbolehkan, sehingga meningkatkan keamanan dan mempermudah proses penggunaan tanpa kunci fisik.



Gambar 2. 3 *ESP32-Cam* ([randomnerdtutorials-com](https://randomnerdtutorials.com), 2025)

2.1.7.3 Sensor *Ultrasonik*

Sensor *Ultrasonik* adalah sensor yang bekerja dengan memancarkan gelombang ultrasonik dan mengukur waktu pantulannya untuk mengetahui jarak objek. Sensor ini sering digunakan untuk mendeteksi keberadaan atau kedekatan benda karena proses pengukurannya cepat dan tidak memerlukan sentuhan fisik (Ananda et al., 2023).

Dalam proyek *Smart Locker* ini, Sensor *Ultrasonik* digunakan untuk mendeteksi apakah terdapat barang di dalam *Locker* atau tidak. Data jarak yang terbaca akan menentukan status *Locker* sebagai kosong atau terisi, kemudian informasi tersebut dikirimkan ke sistem monitoring agar pengguna dan pengelola dapat memantau *Locker* secara real-time. Dengan demikian, sensor ini membantu mencegah barang tertinggal dan mendukung efisiensi pengelolaan *Locker*.

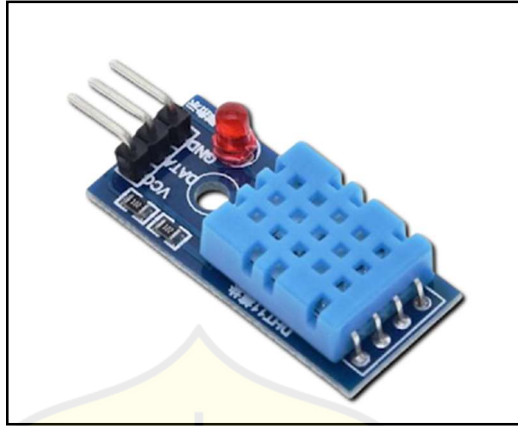


Gambar 2. 4 Sensor *Ultrasonik* (<https://www.arduinoindonesia.id>, 2025)

2.1.7.4 Sensor *DHT11*

Sensor *DHT11* merupakan sensor digital yang digunakan untuk mengukur suhu dan kelembaban udara di lingkungan sekitarnya. Sensor ini memiliki tingkat akurasi yang baik dan mudah diintegrasikan dengan mikrokontroler seperti *ESP32*, sehingga umum digunakan dalam sistem berbasis Internet of Things (IoT) untuk pemantauan kondisi lingkungan (Ananda et al., 2023).

Dalam proyek *Smart Locker* ini, sensor *DHT11* digunakan untuk memantau suhu dan kelembaban di dalam *Locker*, terutama jika digunakan untuk menyimpan barang elektronik atau dokumen penting. Data dari sensor ini dikirim ke sistem monitoring agar pengguna maupun pengelola dapat mengetahui kondisi *Locker* secara real-time. Dengan demikian, sensor *DHT11* membantu menjaga keamanan dan kualitas penyimpanan barang di dalam *Locker*.



Gambar 2. 5 Sensor *DHT11* (<https://www.arduinoindonesia.id>, 2025)

2.1.7.5 Solenoid Lock

Solenoid lock merupakan aktuator pengunci elektronik yang bekerja dengan prinsip elektromagnetik, di mana arus listrik digunakan untuk menggerakkan pin pengunci secara otomatis. Komponen ini banyak digunakan dalam sistem keamanan modern karena mampu membuka dan menutup kunci tanpa interaksi manual (Tawakal & Ramdhani, 2021).

Dalam proyek *Smart Locker* ini, solenoid lock dikendalikan oleh mikrokontroler *ESP32* untuk mengatur mekanisme buka dan tutup *Locker* setelah proses autentikasi *QR Code* atau *RFID* berhasil. Saat pengguna terverifikasi, solenoid lock akan melepaskan pin pengunci agar pintu dapat dibuka, lalu kembali menutup secara otomatis setelah digunakan. Dengan demikian, aktuator ini menjadi komponen utama dalam menjaga keamanan fisik *Locker* secara efisien dan otomatis.



Gambar 2. 6 Solenoid lock (<https://forum.digikey.com>, 2025)

2.1.7.6 Buzzer

Buzzer merupakan komponen elektronik yang menghasilkan suara sebagai bentuk notifikasi atau peringatan. Komponen ini sering digunakan dalam sistem otomatisasi untuk memberikan umpan balik (*feedback*) kepada pengguna terhadap suatu kondisi atau peristiwa tertentu (Ananda et al., 2023).

Dalam proyek *Smart Locker* ini, buzzer berfungsi sebagai indikator suara untuk menandakan hasil proses autentikasi. Ketika pemindaian *QR Code* atau *RFID* berhasil, buzzer akan berbunyi pendek sebagai tanda akses diterima, sedangkan jika gagal, akan mengeluarkan bunyi berbeda sebagai peringatan. Dengan adanya buzzer, pengguna dapat dengan mudah mengetahui status operasi *Locker* tanpa harus melihat tampilan layar.



Gambar 2. 7 Buzzer (<https://www.arduino.biz.id>, 2025)

2.1.7.7 LED 5mm

LED (*Light Emitting Diode*) merupakan komponen elektronika yang dapat memancarkan cahaya ketika dialiri arus listrik. Komponen ini sering digunakan sebagai indikator visual dalam berbagai sistem elektronik karena konsumsi dayanya rendah dan responsnya cepat (Ananda et al., 2023).

Dalam proyek *Smart Locker* ini, LED digunakan sebagai penanda status *Locker*, di mana LED menyala dengan warna tertentu untuk menunjukkan kondisi *Locker*, seperti menyala merah saat sedang digunakan dan hijau ketika tersedia. Indikator visual ini mempermudah pengguna maupun petugas untuk mengenali status setiap *Locker* secara langsung tanpa harus membuka aplikasi monitoring.

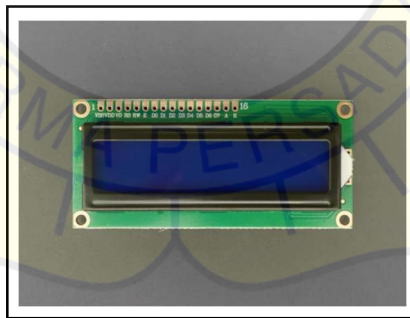


Gambar 2. 8 LED 5mm (<https://id.chiplid-zd.com>, 2025)

2.1.7.8 LCD Display

LCD (*Liquid Crystal Display*) merupakan komponen tampilan yang digunakan untuk menampilkan informasi berupa teks atau angka secara digital. LCD banyak digunakan pada sistem berbasis mikrokontroler karena mudah diintegrasikan dan mampu menampilkan data secara jelas dengan konsumsi daya yang rendah (Suryani et al., 2024).

Dalam proyek *Smart Locker* ini, LCD berfungsi sebagai media informasi bagi pengguna, seperti menampilkan status autentikasi *QR Code* atau *RFID*, nomor *Locker* yang sedang digunakan, serta instruksi penggunaan sistem. Dengan adanya LCD, pengguna dapat menerima umpan balik secara langsung selama proses penggunaan *Locker*, sehingga meningkatkan interaktivitas dan kemudahan dalam pengoperasian.



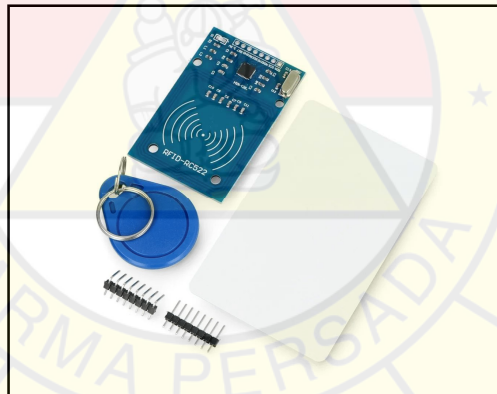
Gambar 2. 9 LCD Display (www-electronicwings-com, 2025)

2.1.7.9 RFID Reader

RFID Reader merupakan perangkat yang digunakan untuk membaca data dari tag atau kartu *RFID* menggunakan gelombang radio. Teknologi ini memungkinkan

proses identifikasi tanpa kontak langsung dan sering digunakan dalam sistem keamanan serta manajemen akses karena efisien dan mudah dioperasikan (Pratama, 2022).

Dalam proyek *Smart Locker* ini, *RFID* Reader berfungsi sebagai metode autentikasi alternatif selain *QR Code*, sehingga pengguna dapat membuka atau menutup *Locker* dengan menggunakan kartu *RFID*. Sistem ini memberikan fleksibilitas bagi pengguna yang tidak ingin menggunakan pemindaian QR melalui smartphone, sekaligus meningkatkan keamanan dengan menghubungkan setiap kartu *RFID* ke identitas pengguna yang terdaftar.



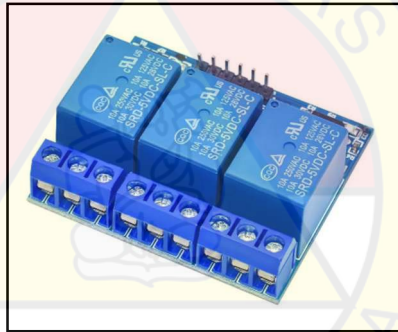
Gambar 2. 10 *RFID* Reader (<https://makerselectronics.com>, 2025)

2.1.7.10 Relay

Relay merupakan komponen elektromekanis yang berfungsi sebagai saklar elektronik untuk menghubungkan atau memutuskan arus listrik pada rangkaian. Komponen ini bekerja dengan memanfaatkan sinyal kontrol bertegangan rendah dari

mikrokontroler untuk mengendalikan beban bertegangan lebih tinggi seperti motor, lampu, atau pengunci elektronik (Tawakal & Ramdhani, 2021).

Dalam proyek *Smart Locker* ini, relay digunakan untuk mengontrol solenoid lock berdasarkan perintah dari mikrokontroler *ESP32*. Ketika autentikasi *QR Code* atau *RFID* berhasil, relay akan mengalirkan arus ke solenoid lock agar pintu *Locker* terbuka. Setelah proses selesai, relay memutuskan arus kembali untuk mengunci *Locker* secara otomatis. Dengan demikian, relay berperan penting sebagai penghubung antara sistem kendali digital dan perangkat pengunci mekanis.



Gambar 2. 11 Relay (<https://store.ichibot.id>, 2025)

2.1.7.11 Power Supply 12V

Power supply 12V merupakan sumber daya listrik yang digunakan untuk menyediakan tegangan dan arus stabil bagi komponen elektronika dalam suatu sistem. Tegangan 12V umum digunakan karena kompatibel dengan berbagai perangkat seperti solenoid, relay, dan modul sensor yang membutuhkan daya lebih besar dibandingkan mikrokontroler (Ananda et al., 2023).

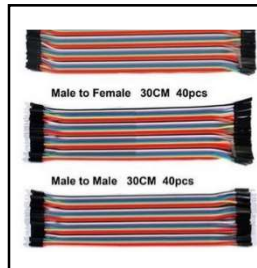
Dalam proyek *Smart Locker* ini, power supply 12V berfungsi sebagai sumber utama untuk mengoperasikan solenoid lock, relay, dan rangkaian elektronik pendukung lainnya, sementara *ESP32* memperoleh daya dari regulator yang menurunkan tegangan ke level 5V atau 3.3V. Dengan penggunaan power supply yang stabil, sistem dapat bekerja secara optimal dan mencegah gangguan pada proses autentikasi maupun penguncian *Locker*.



Gambar 2. 12 Power Supply 12v (<https://1ss-solution.com>, 2025)

2.1.7.12 Kabel Jumper

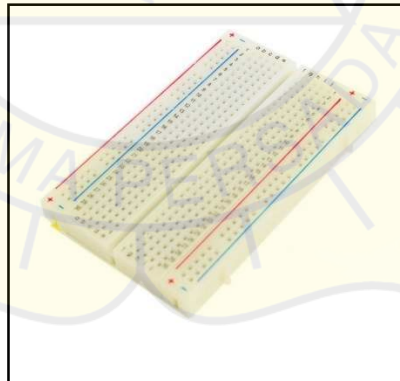
Kabel jumper berfungsi sebagai penghubung antar komponen elektronik pada rangkaian prototipe tanpa perlu disolder (Suryani et al., 2024). Dalam proyek *Smart Locker*, kabel ini digunakan untuk menghubungkan *ESP32* dengan sensor, modul *RFID*, relay, dan aktuator, sehingga sistem dapat berfungsi dengan baik dan mudah dirakit ulang.



Gambar 2. 13 Kabel Jumper (<https://musbikhin.com>, 2025)

2.1.7.13 Breadboard

Breadboard merupakan media perakitan sementara yang digunakan untuk menguji koneksi antar komponen tanpa penyolderan permanen (Ananda et al., 2023). Pada proyek ini, breadboard berperan penting dalam pengujian awal sistem *Smart Locker*, karena memungkinkan proses perbaikan, penambahan, atau penggantian komponen dilakukan secara fleksibel sebelum rangkaian difinalisasi pada papan PCB.



Gambar 2. 14 Breadboard (<https://www.edukasi elektronik a.com>, 2025)

2.1.8 Pemodelan Sistem UML

UML (*Unified Modeling Language*) adalah bahasa pemodelan standar yang digunakan untuk menggambarkan sistem atau aplikasi sebelum dikembangkan. UML membantu pengembang memahami alur, struktur, dan fungsionalitas sistem secara visual. Beberapa jenis sistem UML antara lain :

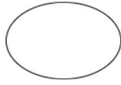
2.1.8.1 Use Case Diagram

Use case Diagram adalah diagram yang menggambarkan tingkah laku secara teratur yang terjadi diantara aktor dan sistem/perangkat. Dalam konteks pengembangan sistem, use case diagram memberikan gambaran umum mengenai fungsionalitas yang disediakan oleh sistem serta bagaimana pengguna (aktor) berinteraksi dengan sistem tersebut (Koç et al., 2021).

Tabel 2. 1 Tipe Relasi pada Use Case Diagram

No	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan use case
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan

			mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (independent).
3	←	<i>Generalization</i>	Hubungan dimana objek anak (descendent) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor)
4	----->	<i>Include</i>	Menspesifikasikan bahwa use case sumber secara eksplisit.
5	←	<i>Extend</i>	menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6	—	<i>Association</i>	menghubungkan antara objek satu dengan objek lainnya.
7	□	<i>System</i>	menspesifikasikan paket yang menampilkan sistem secara terbatas.

8		<i>Use Case</i>	deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah danelemen-elemennya (sinergi).
10		<i>Note</i>	elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

2.1.8.2 Activity Diagram

Activity Diagram merupakan pemodelan yang menggambarkan sebuah sistem kerja dari sebuah objek atau sebuah sistem. Diagram ini digunakan untuk mendeskripsikan proses atau prosedur dalam suatu sistem, baik itu alur bisnis maupun alur operasional suatu aplikasi (la Hnatkowska & Cebinka, 2021).

Tabel 2. 2 Tipe Relasi pada Activity Diagram

No	GAMBAR	NAMA	KETERANGAN
1		<i>Start Point</i>	Titik awal proses. Ditandai dengan lingkaran hitam penuh. Setiap diagram aktivitas selalu dimulai dari titik ini.
2		<i>Activity</i>	Menunjukkan suatu aktivitas atau proses yang dilakukan oleh aktor atau sistem. Contoh: "Kelola Manajemen User", "Sinkronisasi Data", dll.
3		<i>Decision Node</i>	Menandakan titik pengambilan keputusan, di mana alur bercabang berdasarkan kondisi tertentu. Contoh: "Terdapat Data Pegawai?" dengan cabang "Ya/Tidak".
4		<i>Control Flow</i>	Menghubungkan antar elemen (dari satu aktivitas ke aktivitas

			lain atau dari start ke aktivitas). Menunjukkan urutan alur kerja.
5		<i>end state</i>	Titik akhir dari aktivitas dalam sistem. Ditandai sebagai titik di mana proses selesai atau laporan dicetak.

2.1.9 Software Yang Digunakan

Dalam pembuatan aplikasi monitoring *Smart Locker*, terdapat beberapa tools dan *Software* yang digunakan untuk membantu proses perancangan, pengembangan, hingga implementasi sistem.

2.1.9.1 Software

Software atau perangkat lunak adalah kumpulan instruksi, program, dan data yang mengatur operasi perangkat keras serta memungkinkan pengguna berinteraksi dengan sistem secara efisien. Dalam konteks IoT, *Software* mencakup sistem operasi embedded, IDE pemrograman, serta platform cloud untuk pengolahan data *real-time*. *Software* berfungsi sebagai *otak* sistem yang mengintegrasikan input sensor, menjalankan logika kontrol, dan menyajikan output melalui antarmuka web atau *mobile*. Hal ini sesuai dengan penelitian yang menyatakan bahwa perangkat IoT menggabungkan perangkat keras dan perangkat lunak untuk menghubungkan,

berkomunikasi, dan mentransfer data antarperangkat secara otomatis dan efisien dalam ekosistem IoT (Ayu Syahfitri, 2025).

2.1.9.2 Arduino IDE

Arduino IDE (Integrated Development Environment) merupakan perangkat lunak sumber terbuka (*open source*) yang digunakan untuk menulis, mengompilasi, dan mengunggah program ke papan Arduino. *Arduino IDE* mendukung berbagai sistem operasi seperti Windows, macOS, dan Linux, serta menggunakan bahasa pemrograman berbasis C/C++ dengan dukungan *compiler* AVR-GCC dan *framework* Processing. Perangkat lunak ini dirancang untuk menyederhanakan proses pengembangan aplikasi mikrokontroler, mulai dari penulisan kode sumber, proses kompilasi, pengunggahan program ke mikrokontroler, hingga pengujian melalui *serial monitor* (Shakirovich Ismailov Zafar Botirovich Jo, 2022).

2.1.9.3 Android Studio

Android Studio merupakan *Integrated Development Environment* (IDE) resmi yang digunakan untuk pengembangan aplikasi Android. Dikembangkan oleh Google, *Android Studio* menyediakan berbagai alat pendukung seperti *code editor*, *debugger*, *emulator*, serta sistem manajemen proyek yang terintegrasi untuk membantu pengembang dalam membangun aplikasi Android yang efisien dan berkualitas tinggi (Kurnia, 2025).

2.1.9.4 Webview

WebView merupakan komponen pada aplikasi *mobile* Android yang digunakan untuk menampilkan konten berbasis *website* ke dalam aplikasi tanpa harus membuka *browser* eksternal. Dengan memanfaatkan WebView, halaman web yang telah dibuat sebelumnya dapat langsung diintegrasikan ke dalam aplikasi *mobile*, sehingga pengguna dapat mengakses informasi secara lebih praktis dan efisien melalui satu aplikasi (Febiola, 2022).

Menurut Febiola (2022), penggunaan WebView dalam aplikasi Android memberikan kemudahan dalam pengembangan sistem karena perubahan atau pembaruan konten cukup dilakukan pada sisi *website* tanpa perlu melakukan perubahan pada aplikasi secara keseluruhan. Hal ini menjadikan WebView sebagai solusi yang efektif untuk aplikasi berbasis informasi dan monitoring, termasuk pada sistem *Internet of Things* (IoT) yang membutuhkan tampilan data secara *real-time* melalui aplikasi *mobile*.

2.1.9.5 MySQL dan phpMyAdmin

MySQL merupakan salah satu *Relational Database Management System* (RDBMS) *open-source* yang paling banyak digunakan. MySQL memungkinkan pengguna untuk menyimpan dan mengelola data dengan struktur tabel menggunakan bahasa *Structured Query Language* (SQL). phpMyAdmin adalah aplikasi berbasis *web* yang mempermudah pengguna dalam mengelola *Database* MySQL. Dengan antarmuka grafis, phpMyAdmin memungkinkan pengguna untuk menjalankan *query*

SQL, membuat tabel, serta mengelola relasi antar tabel tanpa harus menggunakan *command line*.

Dalam pengembangan sistem informasi akademik, MySQL dan phpMyAdmin terbukti mendukung proses pengelolaan data yang terstruktur dan efisien. Kombinasi ini mempermudah manajemen basis data dengan antarmuka yang ramah pengguna, seperti dijelaskan oleh (Wulandari, 2022) dalam studi implementasi Laravel untuk sistem akademik.

2.1.10 Bahasa Pemrograman

Bahasa pemrograman adalah alat komunikasi antara manusia dan komputer yang berfungsi untuk memberikan perintah dalam pengembangan aplikasi atau sistem perangkat lunak. Bahasa ini digunakan untuk membangun logika program, mengolah data, serta mendukung pengembangan antarmuka (*frontend*) dan sisi server (*backend*) (Fadilah Aulia & Yahfizham, 2024).

2.1.10.1 Hypertext Preprocessor (PHP)

PHP adalah bahasa pemrograman sisi server (server-side) yang digunakan secara luas untuk pengembangan aplikasi *web* dinamis. Sebagai bahasa open-source, PHP memungkinkan integrasi yang mudah dengan berbagai sistem basis data dan memiliki komunitas pengembang yang besar.

Dalam penelitian oleh (Endra et al., 2021), dilakukan analisis perbandingan antara PHP Laravel dan PHP Native dalam pengembangan *website*. Studi ini

menunjukkan bahwa penggunaan Laravel, yang merupakan framework PHP, dapat meningkatkan efisiensi dan struktur dalam pengembangan aplikasi *web* dibandingkan dengan penggunaan PHP secara native

2.1.10.2 Javascript

JavaScript adalah bahasa pemrograman yang awalnya dikembangkan untuk dijalankan di sisi klien (*client-side*) dalam peramban *web*, memungkinkan interaktivitas dan dinamika pada halaman *web*. Namun, seiring perkembangan teknologi, JavaScript kini digunakan secara luas di sisi server melalui *platform* seperti Node.js, serta dalam pengembangan aplikasi mobile dan *Internet of Things (IoT)* (Fadilah Aulia & Yahfizham, 2024).

2.1.10.3 Hypertext Markup Language (HTML)

HTML adalah bahasa dasar untuk membangun struktur halaman *web*, menggunakan tag untuk menandai elemen seperti teks, gambar, tautan, dan formulir. Memungkinkan pembuatan link (*hypertext*) yang menghubungkan satu dokumen dengan dokumen lain membentuk navigasi antarhalaman *web*. Sebagai fondasi *website*, HTML menentukan kerangka konten yang kemudian dapat ditata dengan CSS atau diberi interaktivitas dengan JavaScript (Fadilah Aulia & Yahfizham, 2024).

2.1.10.4 Cascading Style Sheets (CSS)

CSS adalah bahasa *stylesheet* yang digunakan untuk mengatur tampilan halaman *web* yang dibuat dengan HTML. CSS mengendalikan elemen seperti warna, *font*, tata

letak, ukuran gambar, dan spasi, memastikan desain yang seragam dan menarik. CSS memungkinkan pemisahan antara struktur (HTML) dan gaya visual, sehingga mempermudah pengelolaan tampilan *website* (Fadilah Aulia & Yahfizham, 2024).

2.1.11 Database dan DBMS

Database adalah kumpulan data yang terorganisir secara sistematis untuk menyimpan, mengelola, dan mengambil informasi. *Database Management System* (DBMS) adalah perangkat lunak yang digunakan untuk mengelola *Database*. Dalam proyek ini, MySQL digunakan sebagai DBMS untuk mengelola data secara efisien, sedangkan phpMyAdmin digunakan sebagai alat berbasis *web* untuk mempermudah administrasi *Database* MySQL melalui antarmuka grafis (Wulandari, 2022).

2.1.11.1 Database

Database, atau basis data, adalah kumpulan data yang terorganisir secara sistematis dan disimpan secara elektronik dalam sistem komputer. Tujuan utama dari *Database* adalah untuk memudahkan penyimpanan, pengelolaan, dan pengambilan data secara efisien (Wulandari, 2022).

2.1.11.2 Database Management System (DBMS)

Database Management System (DBMS) adalah perangkat lunak yang digunakan untuk membuat, mengelola, dan memanipulasi *Database*. DBMS menyediakan antarmuka antara pengguna dan *Database*, memungkinkan pengguna untuk melakukan

operasi seperti penyimpanan, pembaruan, penghapusan, dan pengambilan data dengan cara yang efisien dan terkontrol (Wulandari, 2022).

2.1.11.3 Relasi Antar Tabel (Database Relationship)

Relasi antar tabel merupakan hubungan logis antara dua atau lebih tabel dalam sistem *database* relasional yang dibangun untuk menghubungkan data secara terstruktur tanpa terjadi redundansi. Relasi ini umumnya dibentuk melalui penggunaan *primary key* sebagai identitas unik pada suatu tabel dan *foreign key* sebagai atribut yang mereferensikan *primary key* pada tabel lain, sehingga tercipta keterkaitan data yang konsisten dan terintegrasi. Penerapan relasi antar tabel bertujuan untuk menjaga integritas data (*data integrity*), meningkatkan efisiensi pengelolaan informasi, serta mendukung proses pengambilan dan pengolahan data secara sistematis dalam suatu sistem informasi (Koç et al., 2021).

2.1.11.3.1 One to One (1:1)

Relasi *one to one* (1:1) merupakan jenis hubungan antar tabel di mana satu data pada tabel pertama hanya berhubungan dengan satu data pada tabel kedua, dan sebaliknya. Relasi ini biasanya diterapkan untuk memisahkan data yang bersifat spesifik atau sensitif agar struktur tabel tetap terorganisir dan tidak terlalu kompleks. Implementasi relasi 1:1 dilakukan dengan menempatkan *foreign key* yang memiliki sifat unik (*unique constraint*) sehingga setiap baris data hanya memiliki satu pasangan

relasi. Model relasi ini mendukung normalisasi data dan menjaga konsistensi struktur basis data relasional (Koç et al., 2021).

2.1.11.3.2 One to Many (1:N)

Relasi *one to many* (1:N) adalah hubungan antar tabel di mana satu data pada tabel utama dapat memiliki banyak data terkait pada tabel lain, tetapi setiap data pada tabel kedua hanya terhubung ke satu data pada tabel utama. Relasi ini merupakan bentuk hubungan yang paling umum digunakan dalam sistem informasi, terutama dalam pencatatan transaksi atau riwayat aktivitas. Implementasinya dilakukan dengan menambahkan *foreign key* pada tabel kedua yang mereferensikan *primary key* pada tabel pertama. Relasi 1:N mendukung efisiensi penyimpanan data dan memungkinkan pengelolaan informasi secara terstruktur tanpa redundansi (Ia Hnatkowska & Cebinka, 2021).

2.1.11.3.3 Many to Many (M:N)

Relasi *many to many* (M:N) adalah hubungan di mana banyak data pada tabel pertama dapat berhubungan dengan banyak data pada tabel kedua. Dalam implementasinya, relasi ini tidak dapat dibentuk secara langsung, melainkan memerlukan tabel penghubung (*junction table* atau *associative table*) yang berisi kombinasi *foreign key* dari kedua tabel yang saling berelasi. Pendekatan ini memungkinkan fleksibilitas dalam pengelolaan data yang kompleks serta menjaga integritas hubungan antar entitas dalam basis data relasional. Relasi M:N banyak

digunakan pada sistem yang memiliki keterkaitan ganda antar entitas dan mendukung desain database yang lebih terstruktur (Rahaman, 2024).

2.1.11.4 Tipe Join pada SQL

Join pada SQL merupakan operasi yang digunakan untuk menggabungkan data dari dua atau lebih tabel berdasarkan kolom yang memiliki hubungan, umumnya melalui *primary key* dan *foreign key*. Penggunaan *join* memungkinkan sistem menampilkan data yang terintegrasi tanpa perlu menyimpan informasi secara berulang dalam satu tabel. Beberapa tipe *join* yang umum digunakan antara lain *INNER JOIN*, yang menampilkan data yang memiliki kecocokan pada kedua tabel; *LEFT JOIN*, yang menampilkan seluruh data pada tabel kiri beserta data yang cocok dari tabel kanan; *RIGHT JOIN*, yang menampilkan seluruh data pada tabel kanan beserta pasangan yang sesuai dari tabel kiri; serta *FULL OUTER JOIN*, yang menampilkan seluruh data dari kedua tabel meskipun tidak memiliki pasangan relasi (meskipun tipe ini tidak didukung secara langsung oleh MySQL). Implementasi *join* sangat penting dalam sistem berbasis database relasional karena mendukung proses pengambilan data yang kompleks dan efisien dalam aplikasi berbasis web maupun IoT (Rahaman, 2024).

2.1.11.4.1 Inner Join

Inner Join merupakan tipe *join* pada SQL yang digunakan untuk menampilkan data dari dua tabel atau lebih berdasarkan kondisi kecocokan pada kolom yang berelasi. Hanya baris data yang memiliki nilai yang sama pada kolom penghubung (*matching*

records) yang akan ditampilkan dalam hasil query, sehingga data yang tidak memiliki pasangan relasi tidak akan muncul. Tipe ini paling umum digunakan dalam sistem basis data relasional karena mampu menghasilkan data yang akurat dan terfilter sesuai kebutuhan relasi antar tabel. Penggunaan *Inner Join* mendukung efisiensi pengolahan data dalam aplikasi berbasis web dan sistem informasi modern (Rahaman, 2024).

2.1.11.4.2 Left Join

Left Join atau *Left Outer Join* adalah tipe *join* yang menampilkan seluruh data dari tabel sebelah kiri dan hanya menampilkan data yang cocok dari tabel sebelah kanan. Jika tidak ditemukan kecocokan, maka nilai pada tabel kanan akan bernilai *NULL*. Tipe ini sering digunakan ketika sistem perlu memastikan bahwa seluruh data utama tetap ditampilkan meskipun tidak memiliki pasangan relasi pada tabel lainnya. Dalam sistem informasi berbasis database relasional, *Left Join* membantu menjaga kelengkapan informasi tanpa menghilangkan data utama (Ia Hnatkowska & Cebinka, 2021).

2.1.11.4.3 Right Join

Right Join atau *Right Outer Join* memiliki konsep yang serupa dengan *Left Join*, namun menampilkan seluruh data dari tabel sebelah kanan dan hanya data yang cocok dari tabel sebelah kiri. Jika tidak terdapat kecocokan, maka kolom dari tabel kiri akan bernilai *NULL*. Tipe ini digunakan ketika tabel kanan dianggap sebagai sumber data utama dalam proses pengambilan informasi. Implementasi *Right Join* membantu dalam

analisis data yang membutuhkan fokus pada tabel referensi tertentu dalam sistem relasional (Koç et al., 2021).

2.1.11.4.4 Full Join

Full Join atau *Full Outer Join* adalah tipe *join* yang menampilkan seluruh data dari kedua tabel, baik yang memiliki kecocokan maupun yang tidak memiliki pasangan relasi. Jika tidak terdapat kecocokan antar tabel, maka nilai pada kolom yang tidak memiliki pasangan akan diisi dengan *NULL*. Meskipun tidak semua DBMS seperti MySQL mendukung *Full Join* secara langsung, konsep ini tetap penting dalam teori basis data karena memberikan gambaran menyeluruh terhadap relasi antar entitas dalam sistem informasi. Penggunaan *Full Join* mendukung analisis data secara komprehensif dalam pengolahan database relasional (Rahaman, 2024).

2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode Prototyping. Metode ini dipilih karena proses pengembangan sistem *Smart Locker* membutuhkan tahapan perancangan dan pengujian secara bertahap untuk memperoleh rancangan yang sesuai dengan kebutuhan pengguna. Metode *prototyping* memungkinkan peneliti untuk membuat model awal sistem yang kemudian diuji, dievaluasi, dan disempurnakan berdasarkan umpan balik pengguna hingga sistem berfungsi sesuai dengan yang diharapkan. Metode ini dinilai tepat karena pengembangan sistem *Smart Locker* berbasis IoT melibatkan integrasi perangkat keras

dan perangkat lunak yang memerlukan penyesuaian berulang selama proses implementasi (munif et al., 2020).

2.2.1 Tahapan Metode Prototyping

Tahapan metode prototyping dalam penelitian ini meliputi:

1. Pengumpulan Kebutuhan Sistem

Pada tahap ini dilakukan identifikasi kebutuhan pengguna dan kebutuhan sistem, seperti kebutuhan fitur autentikasi *QR Code* dan *RFID* , pemantauan status *Locker* secara real-time, dan mekanisme penguncian otomatis. Selain itu, ditentukan juga komponen perangkat keras yang diperlukan seperti *ESP32*, *ESP32-CAM*, Sensor *Ultrasonik*, solenoid lock, buzzer, dan LCD Display.

2. Perancangan Prototipe

Desain sistem dilakukan dengan membuat diagram alur kerja, rancangan rangkaian elektronika, dan rancangan struktur perangkat lunak. Prototipe fisik *Locker* dan kotak kontrol elektronik juga mulai disusun untuk menyesuaikan tata letak komponen.

3. Pembuatan dan Integrasi Sistem

Pada tahap ini dilakukan proses perakitan perangkat keras dan pemrograman *ESP32* serta *ESP32-CAM* untuk mendukung autentikasi *QR Code* dan *RFID* . Sistem

IoT diintegrasikan agar data status *Locker* dapat dikirim dan dipantau melalui aplikasi mobile.

4. Pengujian Prototipe

Prototipe diuji untuk memastikan fungsi autentikasi, pendeteksian barang menggunakan Sensor *Ultrasonik*, proses buka/tutup solenoid lock, serta respon buzzer dan tampilan pada LCD Display. Pengujian dilakukan untuk melihat kecepatan pemindaian *QR Code*, akurasi pembacaan *RFID*, dan kestabilan komunikasi IoT.

5. Evaluasi dan Perbaikan

Hasil pengujian dianalisis untuk melihat apakah sistem telah berjalan sesuai kebutuhan. Jika ditemukan kekurangan, dilakukan perbaikan baik pada perangkat keras maupun perangkat lunak hingga prototipe berfungsi optimal.

2.3 Tinjauan Literatur

Tabel 2. 3 Kajian Penelitian terdahulu

No	Judul Penelitian	Penulis	Dipublikasikan di	Tahun	Klasifikasi Jurnal	Pokok Bahasan
1	Perancangan <i>Smart Locker</i> dengan Implementasi Sistem IoT dan Aplikasi Mobile Android	Fitri Elvira Ananda, dkk.	ELECTRICIAN – Jurnal Rekayasa dan Teknologi Elektro	2023	SINTA 3	Membahas perancangan <i>Smart Locker</i> berbasis IoT dengan autentikasi <i>QR Code</i> serta aplikasi Android untuk kontrol buka/tutup dan monitoring status

						<i>Locker</i> secara real-time.
2	<i>Smart Lock Door Menggunakan Akses e-KTP Berbasis Internet of Things</i>	Moch Iqbal Tawakal & Yudi Ramdhani	JURNAL RESPON SIF	2021	SINTA 4	Menjelaskan penggunaan <i>RFID</i> berbasis e-KTP dan IoT menggunakan NodeMCU untuk sistem kunci pintu otomatis. Sistem mampu melakukan otentikasi kartu e-KTP, mengontrol solenoid lock, serta memberikan notifikasi melalui aplikasi Blynk.
3	Sistem Smart Parking Menggunakan Autentikasi <i>QR Code</i> dan IoT	Suryani, Lindawati, & Fadhli	ELKOMI KA – Jurnal Teknik Elektro	2024	SINTA 2	Mengimplementasikan autentikasi <i>QR Code</i> dan komunikasi IoT untuk verifikasi identitas pengguna slot parkir dan monitoring status parkir secara real-time, konsepnya relevan dengan manajemen akses <i>Smart Locker</i> .
4	Implementasi Sistem Keamanan IoT Berbasis <i>QR Code</i> pada <i>Locker</i> untuk Peningkatan Keamanan dan Aksesibilitas	Wahyu Aulia N. Fitri, Ardian A. D. Saputra, dkk.	Jurnal Elkolind	2024	SINTA 4	Fokus pada penerapan <i>QR Code</i> sebagai metode autentikasi utama dalam sistem <i>Locker</i> untuk meningkatkan keamanan dan aksesibilitas pengguna. Menjelaskan mekanisme pemindaian dan

						integrasi dengan server.
5	Penerapan Radio Frequency Identification Dalam Membangun Sistem Keamanan dan Monitoring Smart Lock Door Berbasis Website	Ridho Syukuryansyah, Didik Setiyadi, Syahbaniar Rofiah	INFOTECH: Journal of Technology Information	2020	SINTA 4	Membahas penerapan <i>RFID</i> dan Arduino untuk sistem keamanan pintu, monitoring akses melalui website, alarm buzzer, dan pencatatan aktivitas masuk–keluar.
6	Implementasi Sistem Smart Locker Berbasis Internet of Things Menggunakan QR Code	Shakirovich Ismailov Zafar Botirovich Jo, Alisher	Jurnal Internasional (TPPP / Teknik Informatika)	2024	Jurnal Internasional	Membahas implementasi Smart Locker berbasis IoT dengan autentikasi QR Code, integrasi mikrokontroler, serta sistem monitoring sebagai dasar pengembangan sistem penyimpanan otomatis.
7	A Review on Internet of Things (IoT) Architecture, Technologies, Future Applications & Challenges	Rahaman, M. M.	Zenodo	2022	Jurnal Internasional	Mengulas arsitektur IoT, teknologi pendukung, tantangan, serta arah pengembangan IoT di masa depan yang relevan sebagai landasan teori sistem Smart Locker berbasis IoT.