JURNAL MEDIA INFORMASI TEKNOLOGI
Volume 3, No. 1, Februari 2026
DOI: https://doi.org/10.69616/mit.v3i1.263

Implementation of Cryptography Using the Rivest Cipher (RC4) Algorithm on *QR Code* Security for E-Locker Verification

Abang Muhammad Sayudi Hamzah*, Uray Ristian, Kasliono
Universitas Tanjungpura, Indonesia.
***** Corresponding Author. E-mail: h1051201087@student.untan.ac.id

Article History
Received:
Dec 13th, 2025

Revised:
Jul 18th, 2026

Accepted:
Jul 22nd, 2026

Keywords
QR Code;
Cryptography;
Rivest Cipher 4;
Universally Unique Indentifer.

ABSTRACT

The Internet of Things (IoT) based E-Locker system with QR Code authentication offers ease of access, but is vulnerable to threats to the security of Universally Unique Identifier (UUID) data on locker access. This study aims to implement cryptography with the Rivest Cipher 4 (RC4) algorithm to secure UUID data on the server and determine the authentication time on the QR Code barcode to the ESP32-CAM. The RC4 algorithm uses a static key for the user. The encryption process is carried out by the Mobile with the KSA, PGRA and Base64 processes to produce ciphertext. The decryption process is carried out by the server through the Base64, KSA and PGRA processes to produce Plaintext. The test results on the RC4 UUID algorithm stored in the database are sent to the Mobile for encryption and decryption processes where QR Code authentication to the ESP32-CAM sends data to the server for data validation if successful it sends back a notification to the Buzzer once, if it fails it sounds twice. The results of the security test, sniffing was successful using the original UUID data converted to a QR Code barcode and the QR Code was authenticated to the ESP32-CAM so that it produced 2 sounds on the buzzer so that the locker access failed. The data used for authentication was 40 data from each user authenticated 10 times, with the lowest authentication time of 2.83 seconds, the longest authentication time of 12.41 seconds, and an average time of 5.42 seconds.

PENDAHULUAN

Loker adalah lemari penyimpanan barang yang umumnya dapat ditemukan di berbagai lokasi seperti perpustakaan, tempat olahraga, tempat wisata dan tempat umum lainnya. Fungsi loker yaitu menyediakan tempat aman bagi barang-barang berharga bagi penggunanya [1]. Saat ini sistem keamanan sudah berkembang dengan menggunakan sistem teknologi, pada mulanya sistem keamanan masih menggunakan kunci konvensional dan dilakukan secara manual. Menggunakan loker yang memiliki tingkat keamanan tinggi untuk menjaga barang berharga. Kunci pintu menjadi dasar keamanan yang penting, salah satu sistem keamanan yang dapat digunakan yaitu menggunakan *QR Code* [2].

QR Code adalah model penyimpanan data yang memiliki bentuk martiks dua dimensi yang berbentuk vertical dan horizontal. *QR Code* memiliki kemampuan untuk menyimpan informasi lebih baik dan berkapasitas lebih besar dibanding barcode yang merupakan versi lampaunnya [3]. Pada loker berbasis *QR Code*, UUID digunakan untuk menghasilkan nilai acak yang terdiri dari 32 bilangan hexadecimal yang berfungsi sebagai tanda identifikasi pengguna dalam mengakses loker. UUID dapat diakses jika kode dari pengguna diketahui orang lain dan dapat diakses melalui website

yang dibuat sehingga pengguna lain hanya memasukan kode UUID ke website untuk mengakses loker [4].

Algoritma kriptografi dari metode RC4 merupakan algoritma yang bersifat *stream cipher* dengan sistem keamanan berorientasi pada penyandian satu *bit/Byte* data. RC4 memiliki dua bagian, yaitu: *key setup* atau *key scheduling algorihms* (KSA) dan *stream generation* atau *pseudo-random generation algorithms* (PRGA), Pada *Stream Generation* nilai *pseudorandom* yang dioperasikan XOR dapat menghasilkan ciphertext atau sebaliknya menghasilkan *Plaintext*. sedangkan pada Key Setup terdiri dari 3 proses seperti Inisialisasi S-Box, Menyimpan *key* dalam *Key Byte* Array, Permutasi pada S-Box [5].

Penelitian lainnya berjudul "Enkripsi Data *QR Code* Menggunakan Metode RC4 pada Aplikasi Presensi Jurusan Teknik Informatika Universitas Halu Oleo" [6]. Penelitian ini menghasilkan pada metode RC4 berhasil mengimplementasikan pada data *QR Code* dan menghasilkan kecepatan scanning dari setiap *QR Code* memiliki kisaran 6 sampai 5 detik untuk setiap kelas. *QR Code* yang sudah ditanamkan metode RC4 hanya dapat didekripsi pada sistem yang digunakan, sehingga meminimalisir kecurangan dalam proses presensi mahasiswa.

Penelitian lainnya berjudul "Aplikasi Kehadiran Siswa Berbasis Android menggunakan *QR Code* pada Bimbel *Excellent Institute*" [7]. Penelitian ini menghasilkan sistem kehadiran dalam melakukan pemindaian *QR Code* yang didalamnya menggunakan algortima RC4. RC4 merupakan salah satu jenis kode yang dioperasikan nilai enkripsi dilakukan per-*Byte* dalam sekali operasi. Penelitian ini juga menggunakan black-box testing untuk menguji *user* interface yang dimana memudahkan untuk pengguna mengunakan. Penelitian ini juga menciptakan aplikasi kehadiran yang memudahkan guru dan murid dalam melakukan kehadiran dikelas.

Penelitian terkait yang mengangkat tema tentang verifikasi E-Locker dengan *QR Code* berjudul "Implementasi Smart Helmet Cabinet pada Penyimpanan Helm Berbasis Mobile *QR Code*" [4]. Penelitian ini menghasilkan teknologi *QR Code* sebagai autentikasi akses pengguna, waktu delay yang dihasilkan dalam mengakses loker sekitar 4.73 detik dan autentikasi *QR Code* dengan waktu 5.99 detik. Sistem ini berhasil mengidentifikasi pengguna dengan menggunakan *QR Code* sebagai autentikasi akses.

Berdasarkan permasalahan dan penelitian terkait yang pernah dilakukan maka akan dilakukan penelitian dengan judul "Implementasi Kriptografi Menggunakan Algoritma Rivest Cipher (RC4) pada Keamanan *QR Code* untuk Verifikasi E-Locker". Penelitian ini akan menghasilkan sistem keamanan pada Loker dengan menerapkan metode Algoritma Rivest Cipher (RC4) pada enkripsi dan deksripsi *universally unique identifier* (UUID), sistem yang dibangun akan memiliki autentikasi akses keamanan yang baik dan mengetahui *response time* autentikasi dari *QR Code* ke ESP32-CAM saat proses verifikasi akses E-Locker.

METODE

Metodologi penelitian ini mencakup beberapa langkah, terdiri dari yaitu Studi Literatur, Analisis Kebutuhan, Perancangan Sistem, Implementasi Sistem, Pengujian Sistem, Data Hasil Pengujian, Kesimpulan dan Saran.

Studi Literatur

Pada tahap studi literatur tahapan mencari informasi yang mendukung penelitian untuk mengetahui dasar teori. Literatur yang digunakan berupa jurnal penelitian ilmiah yang pernah dilakukan, buku, dan publikasi lain yang berkaitan dengan masalah serta tujuan penelitian. Berikut informasi yang digunakan dari jurnal terkait dengan penelitian yang dilakukan.

ESP32-CAM

ESP32-CAM merupakan platform yang memiliki pemantauan secara *real-time* dengan adanya kamera dan modul *WiFi* yang ada di papan elektronik nya. Dalam pengaturan pada ESP32-CAM dibutuhkan FTDI USB to TTL yang dimana untuk menghubungkan modul kamera dan perangkat personal seperti komputer atau laptop. Untuk memprogram ke ESP32-CAM memperlukan sebuah aplikasi *opensource* yang didalamnya untuk mengupload program ke modul ESP32-CAM tersebut menggunakan Arduino IDE [8]. Dalam penelitian ini menggunakan ESP32-CAM yang dimana

sangat signifikan dalam mengetahui dan melakukan pemindahan *QR Code*. ESP32-CAM menerima data yang dimana akan melewati proses autentikasi saat memeriksa akses ke loker. ESP32-CAM dapat dilihat pada Gambar 1.

Gambar 1. ESP32-CAM

NodeMCU ESP32

NodeMCU ESP32 merupakan mikrokontroler berbasis chip yang mempunyai kemampuan untuk koneksi internet (*WiFi*). Kontroller ini juga memiliki beberapa pin I/O untuk mengembang sebuah aplikasi monitoring dan control dalam proyek IoT. NodeMCU ESP32 memiliki kesamaan dengan Arduino UNO sehingga memprogram menggunakan Arduino IDE. Fisik pada NodeMCU ESP32 juga dilengkapi port USB, memudahkan untuk proses pemograman. NodeMCU ESP32 juga merupakan modul pengembangan dari platfrom IoT (*Internet of Things*). Fungsi modul sama dengan platform modul Arduino, sehingga fokus pada koneksi internet [8]. Dalam Penelitian ini menggunakan NodeMCU ESP32 yang memiliki dua inti yang dimana mengelolah data yang lebih cepat dan efesien. NodeMCU ESP32 digunakan untuk mengendalikan perangkat pada loker dan mentransfer data dalam jaringan sistem. NodeMCU ESP32 dapat dilihat pada Gambar 2.

Gambar 2. NodeMCU-ESP32

Buzzer

Buzzer merupakan bagian elektronik dimana kegunaan nya untuk mengubah energi listrik menjadi getar atau suara. Buzzer juga memiliki prinsip kerja seperti *loudspeaker*, dalam kerja buzzer meliputi gelombang pada diafragma dan dialirkan arus listrik yang disebut electromagnet, gelombang tersebut juga akan tertarik dari luar ataupun dalam. Pada polaritas magnet memiliki gerakan gelombang secara acak yang akan membuat bergetar udara dan menghasilkan suara. Buzzer juga sering digunakan untuk indikator dalam proses terjadi selesai maupun terjadi kegagalan [9]. Dalam penelitian ini menggunakan buzzer untuk notifikasi bunyi saat berhasil atau gagal pada saat proses autentikasi akses loker. Buzzer dapat dilihat pada Gambar 3.

Gambar 3. Buzzer

Modul Relay 4 Channel

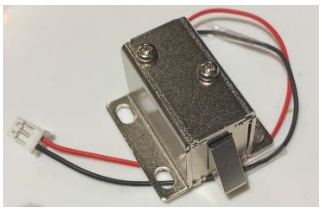
Modul Relay 4 Channel merupakan saklar (*switch*) yang memiliki operasi secara listrik dan komponen Electromechanical terdiri dari dua bagian yaitu Electromagnet (*Coil*) dan Mekanikal (kontak Saklar/*Switch*). Relay mempunyai prinsip elektromagnetik dalam menggerakan kontak saklar sehingga arus listrik menjadi kecil (*low power*) dan arus listrik menjadi tegangan lebih tinggi. Relay juga dapat memutuskan dan menghubungkan supply ke alat listrik lainnya. Cara kerja relay

yaitu memberikann tegangan pada kaki 1 dan kaki ground pada kaki 2 relay maka akan secara otomatis kaki CO (*Change Over*) pada relay akan berpindah dari NC (*Normally Close*) ke NO (*Normally Open*) [10]. Dalam penelitian ini menggunakan modul relay dimana Ketika diaktifkan akan memberikan sinyal listrik pada kumparan elektromagnetik, dimana kumparan tersebut akan menghasilkan medan magnet yang memindahkan kontak saklar dari posisi awal. Modul relay digunakan untuk kontrol rangkaian elektronik pada sistem atau kendali pada loker. Modul Relay 4 Channel dapat dilihat Gambar 4.

Gambar 4. Modul Relay 4 Channel

Solenoid Door Lock

Solenoid Door Lock merupakan akuator untuk menggerakan tuas secara linier. Solenoid Door Lock memiliki fungsi khusus untuk mengunci pintu elektronik. Solenoid Door Lock memiliki di acara kerja yaitu *Normal Close* (NC) dan *Normally Open* (NO). Cara kerja solenoid NC jika diberikan tegangan, maka solenoid NO kebalikan dari solenoid NC. Solenoid Door Lock juga membutuhkan input tegangan kerja 12V DC akan tetapi Solenoid Door Lock juga membutukan input tegangan output dari pin IC digital [11]. Dalam penelitian ini menggunakan solenoid door lock untuk melakukan penguncian dan pembukaan pada loker otomatis, sesuai dengan hak akses pada pengguna. Solenoid Door Lock dapat dilihat Gambar 5.



Gambar 5. Solenoid Door Lock

Kriptografi

Kriptografi merupakan metode enkripsi yang menggunakan kertas dan pensil, atau mungkin dengan bantuan alat mekanik sederhana, dikenal sebagai kriptografi klasik. Secara umum, algoritma kriptografi klasik dikelompokkan menjadi dua kategori, yaitu algoritma transposisi (*transposition cipher*) dan algoritma substitusi (*substitution cipher*). Cipher transposisi mengubah susunan huruf-huruf dalam pesan, sementara cipher substitusi mengganti setiap huruf atau kelompok huruf dengan huruf atau kelompok huruf lain [12].

QR Code

QR Code merupakan gambar berbentuk kisi dua dimensi yang dapat menyimpan informasi di dalamnya. *QR Code* merupakan pengembangan lebih lanjut dari identifikasi standar berupa kode batang (barcode). Karena Kode QR disusun dalam jaringan dua dimensi, maka penyimpanan informasi dilakukan baik secara vertikal maupun horizontal [13]. Dalam sistem E-Locker, *QR Code* berperan sebagai kunci fisik dalam tingkat akses privasi pada loker. *QR Code* yang digunakan mengandung *string* berupa UUID (*Universally Unique Identifier*) sebagai identifikasi unik bagi setiap pengguna. Setiap pengguna diberikan *QR Code* yang berbeda keunikannya. *QR Code* bersifat dinamis dengan mengalami perubahan dengan seiringnya aktivitas pengguna, seperti pada penambahan akses atau menyimpan sementara barang pada loker. Dinamisme ini melakukan pembaharuan UUID dan *QR Code*. *QR Code* dapat dilihat Gambar 6.

Gambar 6. *QR Code*

Rivest Cipher 4

Rivest Cipher 4 merupakan algortima kunci simetris dengan bentuk *stream cipher*, yaitu melakukan proses unit atau input data. Unit atau data umumnya sebuah *Byte* atau kadang-kadang bit. Algoritma ini juga tidak harus menunggu sejumlah input data tertentu sebelum proses, atau menambahkan *Byte* untuk di enkripsi. Algoritma ini ditemukan pada tahun 1987 oleh Ronald Rivest dan menjadi sebuah simbol untuk keamanan RSA. RC4 yaitu enkripsi *stream simetric proprietary* yang dibuat oleh RSA data *Security Inc* (RSADSI) [14]. RC4 memiliki dua bagian, yaitu: *key scheduling algorihms* (KSA) dan *pseudo-random generation algorithms* (PRGA). Berikut ini akan dijelaskan tentang kedua algoritma tersebut.

Key-Scheduling Algorithm (KSA) digunakan untuk menginisialisasi urutan larit "S". Panjang kunci didefinisikan sebagai jumlah *Byte* dalam kunci, dengan rentang kunci 1 hingga 256, dimana 5 hungga 16, dan Panjang kunci 40 hingga 128 bit.pertam larik identitas penerus "S" diinisialisasi. Kemudian "S" diproses dengan cara yang sama seperti PRGA dengan 256 iterasi, akan tetapi pada saat yang sama digabungkan dalam *Byte* kunci. Berikut merupakan algoritma KSA [6]:

S[i] = i (untuk i = 0…255) j = (j + S[i] + K[i mod key_lenght]) mod 256 swap (S[i], S[j])	(1)

Pseudo-Random Generation Algorithm (PRGA) memodifikasi *state* dan output sebuah *Byte* dari *key-stream*. Dikarenakan penting banyaknya dibutuhkan iterasi. Dalam setiap iterasi, PRGA menginkremen i, menambahkan nilai S yang ditunjuk oleh i sampai j, kemudian menukar nilai S[i] dan S[j], lalu mengembalikan elemen dari S dilokasi S[i] + S[j] (modulo 256). Setiap elemen S ditukarkan dengan elemen lainnya paling tidak satu kali setiap 256 iterasi. Berikut merupakan algoritma PGRA [6]:

i = (i + 1) mod 256 j = (j + S[i]) mod 256 swap (S[i], S[j]) t = (S[i] + S[j]) mod 256 k = S[t]	(2)

Base64

Base64 merupakan salah satu algoritma yang digunakan untuk mengubah dan mengembalikan data ke dalam format ASCII, dengan dasar bilangan basis 64. Metode ini digunakan untuk mengenkripsi data biner menjadi bentuk yang dapat dibaca dan diproses [15]. Dalam sistem E-Locker ini setelah melakukan proses algoritma RC4 lanjut menggunakan *Base64* untuk saat mengirim URL data tidak terjadi kegagalan dan mengubah dari data desimal RC4 menjadi karakter yang melalui proses *Base64*. Kode indeks *Base64* yang perlu diubah seperti simbol '+' diubah menjadi '-' dan '/' diubah menjadi '_'.

Analisis Kebutuhan

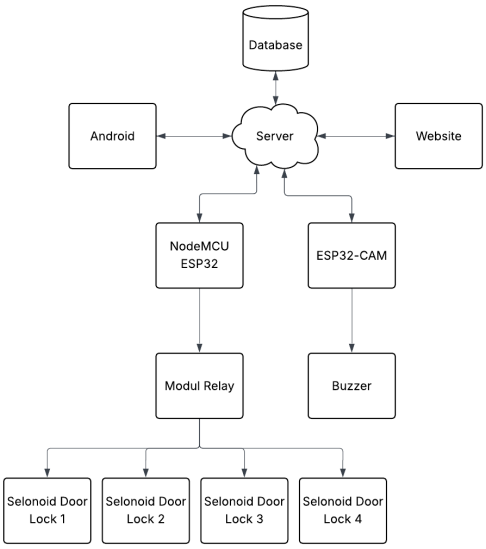
Analisi Kebutuhan yaitu tahapan menganalisa kebutuhan dari sistem yang akan digunakan pada penelitian. Ada dua jenis kebutuhan yang meliputi kebutuhan perangkat keras yaitu ESP32-CAM, NodeMCU ESP32, Buzzer, Modul Relay 4 Channel, Solenoid Door Lock dan kebutuhan perangkat lunak yaitu Arduino IDE, Visual Code Studio, Laravel, Flutter, MySQL dan Hypertext Preprocessor (PHP).

Perancangan Sistem

Perancangan sistem menjelaskan hubungan antara perancangan perangkat keras dan perancangan perangkat lunak yang dibangun. Perancangan perangkat keras terdiri dari NodeMCU ESP32 sebagai pengendali modul relay yang terhubung dengan selonoid door lock pada setiap pintu loker dan pemindaian *QR Code* melalui modul ESP32-CAM serta memberikan sinyal notifikasi ke buzzer. Perancangan Sistem dapat dilihat pada Gambar 7.

Berdasarkan Gambar 7. dapat dijabarkan komponen-komponen rancangan sistem E-Locker berbasis Mobile *QR Code* sebagai berikut:

- NodeMCU ESP32 berfungsi sebagai mikrokontroler dalam mengendalikan beberapa komponen lainnya yang terhubung pada loker.
- Modul ESP32-CAM berfungsi sebagai pemindai *QR Code* serta mengirimkan hasil *QR Code* ke server untuk autentikasi data.
- Modul Relay berfungsi sebagai pemutus dan penghubung dari komponen solenoid door lock pada sistem pintu E-Locker.
- Solenoid Door Lock berfungsi sebagai bagian bagian dasar dalam mengontrol proses penguncian dari setiap pintu loker. Solenoid Door Lock dikendalikan dari NodeMCU ESP32 untuk membuka dan mengunci setiap pintu loker dengan sesuai perintah yang diterima dari sistem.
- Buzzer berfungsi untuk memberitahu mengenai informasi melalui bunyi saat pemindaian berhasil atau gagal pada ESP32-CAM.

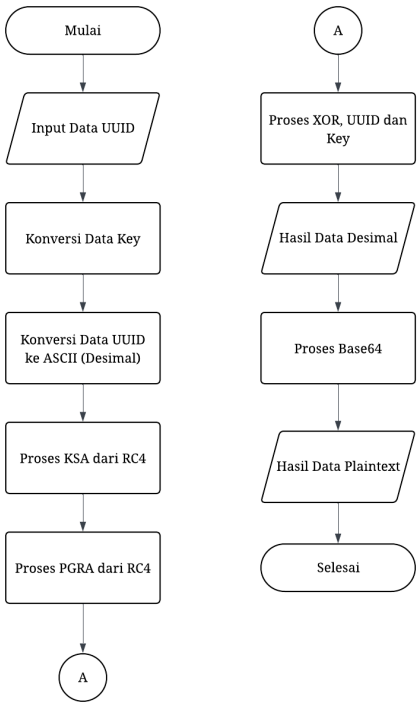


Gambar 7. Perancangan Sistem *E-Locker*

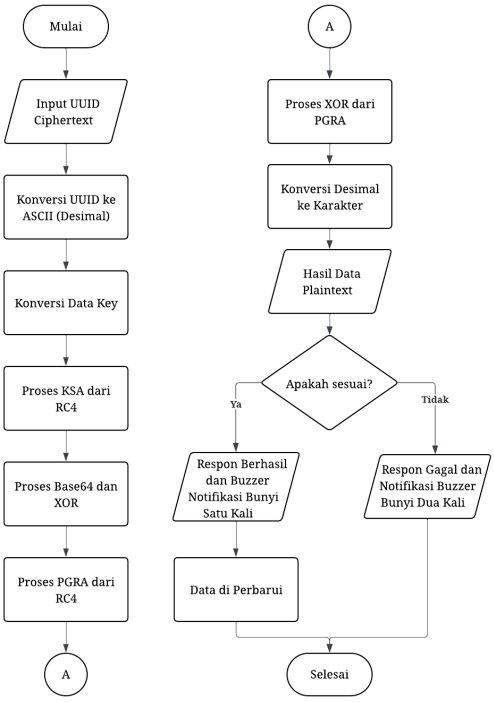
Perancangan proses enkripsi *QR Code* di Aplikasi Mobile pada penelitian ini melakukan proses enkripsi dengan server mengirimkan data ke aplikasi mobile. Aplikasi mobile akan menerima data asli UUID dan key, pada data UUID akan di konversi ke ASCII terlebih dahulu. Setelah itu, melakukan proses dari RC4 yaitu proses *Key Sceduling Algorithm* (KSA) terlebih dahulu dan dilanjutkan dengan proses *Pseudo-Random Generation Algorithm* (PGRA). Setelah melewati dari dua proses tersebut maka mendapatkan hasil dengan bilangan desimal, yang dimana bilangan tersebut akan di konversi kembali ke proses *Base64* dan *Base Encode* yang dimana untuk merubah beberapa tanda seperti (+ menjadi -) dan (/ menjadi _), tanda tersebut digunakan untuk hasil bisa disimpan di URL tanpa *error*. Setelah mendapatkan *string* yang sudah di enkripsi maka data tersebut akan di konversi menjadi balok barcode yaitu *QR Code* Enkripsi. Perancangan proses enkripsi dapat dilihat pada Gambar 8.

Adapun Perancangan proses dekripsi *QR Code* di Server pada penelitian ini melakukan proses dekripsi dengan menerima data autentikasi pada *QR Code*. *QR Code* dikonversi ke string UUID enkripsi dan menerima *key* pada pengguna. Setelah itu, dilakukan proses *Base64* dan XOR yang akan menghasilkan bilangan desimal sebelum lanjut ke proses RC 4 yaitu proses *Key*

Sceduling Algorithm (KSA) dan *Pseudo-Random Generation Algorithm* (PGRA). Jika sudah maka menghasil data UUID yang akan melewati perbandingan untuk autentikasi berhasil atau tidak. Jika berhasil autentikasi data pada database maka, data akan merespon balik ke buzzer yang dimana memberikan notifikasi bunyi sekali dan data UUID akan melakukan pembaruan terbaru. Begitu juga sebaliknya, jika gagal autentikasi data pada database maka data akan merespon balik ke buzzer yang dimana memberikan notifikasi bunyi sebanyak dua kali dan data UUID akan melakukan proses pembaruan terbaru kembali. Perancangan proses dekripsi dapat dilihat pada Gambar 9.



Gambar 8. Perancangan proses enkripsi



Gambar 9. Perancangan proses dekripsi

Adapun proses perhitungan dari algoritma RC4 algoritma kriptografi yang melibatkan dua proses yaitu, KSA untuk inisialisasi *state* array dengan ukuran 256 elemen (0 - 255) yang menggunakan kunci dan PGRA untuk menggenerasi key stream yang akan dipakai untuk XOR dengan data. Dalam sistem E-Locker proses enkripsi berada pada aplikasi mobile. Tujuannya adalah menghasilkan data UUID yang di proses enkripsi dengan algortima RC4. Mengubah karakter ke ASCII bilangan desimal, karakter *key* yang digunakan berjumlah 18 karakter dan UUID yang digunakan berjumlah 32 karakter. Adapun karakter ASCII yang digunakan sebanyak 256 karakter. Berikut konversi data key pada 18 karakter to desimal. Konversi *Plaintext* ke ASCII dapat dilihat pada Tabel 1.

Key: *user*20250424120325

Tabel 1. Konversi *Plaintext* ke ASCII (Desimal)

NO.	*Plaintext* Key	ASCII (Desimal)
1.	u	117
2.	s	115
3.	e	101
4.	r	114
5.	2	50

Berikut karakter *Plaintext* pada UUID yang di konversi berjumlah 32 karakter ke desimal. Konversi UUID ke ASCII dapat dilihat pada Tabel 2.

UUID: 9f1c1a88761d499496d3e92991e3f4b3

Tabel 2. Konversi UUID ke ASCII (Desimal)

NO.	*Plaintext* UUID	ASCII (Desimal)
1.	9	117
2.	f	115
3.	1	101
4.	c	114
5.	1	50

Adapun perhitungan KSA, yaitu proses inisialisasi state array s berukuran 256 dimana elemen (0 – 255) dengan menggunakan kunci. Menghasilkan nilai state array s yang sudah di acak oleh kunci. Berikut perhitungan KSA dapat dilihat Tabel 3.

Tabel 3. Perhitungan KSA

Perhitungan KSA
i = 1
j = (117 + S[1] + K[115]) mod 256
j = (117 + 1 + 115) = 233
swap(S[1], S[233])

Adapun perhitungan PGRA, yaitu proses generasi *key-stream*, yang dimana *key-stream* akan dilakukan proses XOR dengan *Plaintext* (untuk enkripsi). Berikut perhitungan PGRA dilihat Tabel 4.

Tabel 4. Perhitungan PGRA

Perhitungan PGRA
i = 0
i = (0 + 1) mod 256
i = 1
j = (0 + S[1]) mod 256 = 203
sebelum swap: S[1] = 203, S[203] = 233
sesudah swap: S[1] = 233, S[203] = 203
swap(S[1], S[203])
k = (S[1] + S[203]) mod 256
k = (233 + 203) mod 256 = 180
k = S[180] = 149
c = 57 $\oplus$ 149
P = 57 (desimal) = 0011 1001 (biner 8-bit)
K = 149 (desimal) = 1001 0101 (biner 8-bit)
c = 1010 1100
c = 172 (Hasil akhir)

Adapun perhitungan *Base64* merupakan untuk mengubah bilangan desimal menjadi *Byte* biner. Berikut perhitungan *Base64* dilihat Tabel 5.

Tabel 5. Perhitungan *Base64*

Perhitungan *Base64*
Decimal array: 172, 201, 104,
Lanjut mengubah ke Biner dari 3*Byte* yang dimana total ada 24bit
172 = 10101100
201 = 11001001
104 = 01101000

Lanjut mengubah menjadi 1*Byte* berisi 6bit lalu dilanjutkan dengan mengubah ke desimal kembali. 101011 = 43 = r 001100 = 12 = M 100101 = 37 = l 101000 = 40 = o

Setelah itu tahap terakhir mengubah desimal ke karakter *Base64* dari hasil desimal tersebut menjadi karakter ’rMlo’.

Adapun proses dekripsi yaitu mengembalikan data dari *ciphertext* ke *Plaintext* atau mengembalikan data enkripsi ke data asli. Dekripsi dilakukan hampir sama seperti proses dekripsi yang dimana melalui proses KSA dan PGRA setelah itu melakukan XOR dan Proses *Base64*. Pada data KSA dan PGRA sama seperti pada proses enkripsi. Berikut perhitungan *Base64* dekripsi dilihat Tabel 6.

Tabel 6. Perhitungan *Base64* Dekripsi

Perhitungan *Base64*
Data Ciphertext: rMloAvLl8MAY4NbixHt6dE1AUZOfUeGwpk X4I9TETBo=
r = 43 = 101011 M = 12 = 001100 1 = 37 = 100101 o = 10 = 101000
Setelah itu lanjut dengan mengubah menjadi 1*Byte* berisi 8bit dan mengubah ke ASCII desimal.
Biner = 10101100 11001001 01101000
10101100 = 172 11001001 = 201 01101000 = 104

Selanjutnya dikarenakan KSA dan PGRA tidak ada perubahan dari enkripsi sehingga tidak ditampilkan. Lanjut dengan XOR pada PGRA dimana nilai yang dihasilkan sama seperti di proses enkripsi, dimana PGRA menghasilkan nilai k = S[180] = 149. Berikut lanjutan XOR dari PGRA dapat dlihat Tabel 7.

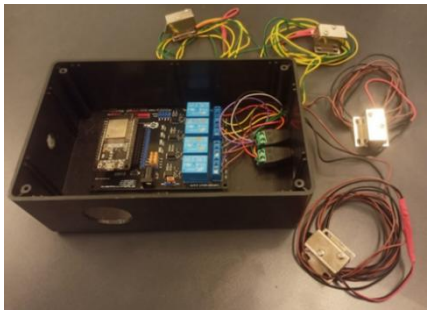
Tabel 7. Perhitungan XOR di PGRA

Perhitungan XOR di PGRA
c = 172 ⊕ 149
P = 172 (desimal) = 1010 1100 (biner 8-bit)
K = 149 (desimal) = 1001 0101 (biner 8-bit)
c = 0011 1001
c = 57 (Hasil akhir)

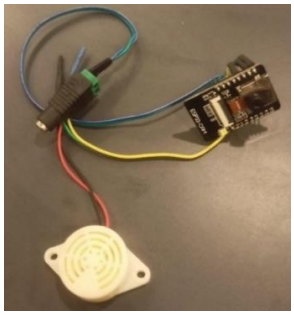
Setelah hasil akhir didapatkan bilangan desimal di ubah menjadi karakter, pada nilai 57 diubah menjadi ASCII karakter ‘9’. Nilai yang didapatkan sesuai dengan nilai awal pada UUID 9f1c dan seterusnya.

Implementasi Sistem

Pada tahapan implementasi memiliki dua tahapan yaitu implementasi perangkat keras dan implementasi perangkat lunak. Pada implementasi perangkat keras dilakukan dengan memastikan sistem pada E-Locker berbasis *Internet of Things* yang berfungsi sesuai dengan desain yang sudah direncanakan. Sistem terdiri dari ESP32-CAM untuk kendali Buzzer dan pemindaian *QR Code*, NodeMCU ESP32 sebagai kendali utama mengontrol Relay Solenoid Door Lock.



Gambar 10. Implementasi sistem dalam mengontrol Modul Relay dan Solenoid Door Lock



Gambar 11. Implementasi Sistem Pemindaian *QR Code*

Implementasi sistem dalam mengontrol Relay dan Solenoid Door Lock pada tahap ini sistem penggunaan mikrokontroler NodeMCU ESP32 yang diintegrasikan dengan modul relay 4 channel, serta Solenoid Door Lock yang ditempelkan dari setiap pintu loker. Pada NodeMCU ESP32 berperan sebagai pengendali utama dalam mengontrol operasi buka dan tutup pada keempat pintu loker. Implementasi sistem dalam mengontrol Modul Relay dan Solenoid Door Lock dapat dilihat Gambar 10.

Implementasi sistem pemindaian *QR Code* pada tahap ini sistem mikrokontroller ESP32-CAM dan buzzer. Mikrokontroler ESP32-CAM berperan sebagai proses dalam pemindaian *QR Code* menggunakan kamera yang ada pada perangkat tersebut. Fungsinya yaitu mendeteksi dan memproses informasi yang terdapat pada *QR Code* yang diarahkan ke pandangan kamera. Sebelum itu *QR Code* yang digunakan dan dibaca sudah melalui proses enkripsi dengan metode RC4 sehingga *QR Code* yang dibaca *QR Code* enkripsi. Proses pemindaian *QR Code* oleh ESP32-CAM akan menghasilkan urutan karakter dalam bentuk string. Hasil pada pemindaian string pada *QR Code*, akan dilanjutkan dengan mengirim hasil pemindaian yang tersebut ke server guna untuk melewati proses autentikasi akses pada loker. Server akan menerima data melakukan proses dekripsi, setelah melakukan proses dekripsi dan memberikan respon balik kepada ESP32-CAM untuk mengidentifikasi apakah proses berhasil atau gagal. buzzer adalah elemen lain yang digunakan untuk memberikan notifikasi kepada pengguna saat melakukan autentikasi *QR Code* ke ESP32-CAM. Ketika autentikasi berjalan dengan sukses, buzzer akan mengeluarkan bunyi sebanyak satu kali. Begitu juga sebaliknya, jika proses autentikasi tidak berjalan dengan baik atau gagal, buzzer akan mengeluarkan bunyi sebanyak dua kali. Implementasi Sistem Pemindaian *QR Code* dapat dilihat Gambar 11.

Implementasi perangkat lunak merupakan langkah-langkah penyusunan komponen perangkat keras yang telah dirancang pada bab perancangan. Implementasi perangkat lunak terdiri dari implementasi antarmuka website yang dimana sebagai kontrol utama untuk admin dan implementasi antarmuka aplikasi mobile sebagai pengguna untuk mengakses loker menggunakan barcode *QR Code*.

Gambar 12. Implementasi Antarmuka Website Admin

Gambar 13. Implementasi Antarmuka Aplikasi Mobile Pengguna

Implementasi antarmuka website pada tampilan dashboard adalah tampilan awal yang diakses oleh admin setelah berhasil melakukan login pada website. Pada halaman dashboard, terdapat beberapa informasi seperti hak akses, kontrol manual dan status loker. Pada hak akses dapat

mengetahui *user* sudah diberikan akses atau belum pada loker, kontrol manual admin dapat melakukan akses langsung dengan membuka dan menutup loker tanpa melakukan *QR Code* dan Status loker memberikan informasi jumlah *user*, jumlah loker kosong dan penggunaan harian pada loker. Implementasi Antarmuka Website Admin dapat dilihat Gambar 12.

Implememtasi antarmuka aplikasi mobile pada tampilan dashboard aplikasi menampilkan informasi terkait dengan loker yang dimiliki oleh *user*. Halaman ini juga memiliki tombol untuk menampilkan *QR Code* dan menampilkan data profil *user*. Selain itu tombol scan *QR Code* digunakan untuk autentikasi penggunaan loker dan tombol akhiri akses untuk mengakhiri akses pada loker. Implementasi Antarmuka Aplikasi Mobile Pengguna dapat dilihat Gambar 13.

Pengujian Sistem

Pengujian Rivest Cipher 4 dalam sistem E-Locker diterapkan untuk mengamankan data UUID pada *QR Code* dengan melalui dua jalur utama, yaitu pengiriman UUID dari *website* admin ke aplikasi *user*, pengiriman hasil pemindaian *QR Code* dari ESP32-CAM ke server. Pengiriman data UUID *user* dari website yang dibuatkan oleh admin sehingga data tersebut dikirim ke aplikasi mobile untuk *user* dapat menggunakan data tersebut. Tujuan ini untuk mengimplementasikan proses enkripsi yang ada di aplikasi mobile. Setelah pengguna menerima data dan mengakses aplikasi yang sudah disediakan, data UUID asli dan role *user* sebagai Key untuk proses enkripsi. Seperti pada Perancangan Proses Enkripsi *QR Code* di Aplikasi Mobile, data UUID asli dan kunci di konversi terlebih dahulu ke ASCII. Setelah itu data akan melakukan proses KSA yang dimana dilakukan iterasi sebanyak 256 kali. Jika sudah melakukan iterasi KSA 256 kali maka mendapatkan hasil dengan Panjang “S i” 256. Setelah mendapatkan nilai dari KSA dilanjutkan dengan proses PGRA yaitu proses generasi *key-stream*, yang dimana *key-stream* akan dilakukan proses XOR dengan *Plaintext* (untuk enkripsi). Setelah mendapatkan nilai hasil akhir yaitu nilai decimal, dikarenakan pada RC4 hasil akhir desimal sulit untuk di konversi ke karakter dan menjadi barcode *QR Code*. Maka dengan ada tambahan menggunakan *Base64* untuk mempermudah konversi ke karakter. Menggunakan *Base64* untuk saat mengirim URL data tidak terjadi kegagalan dan mengubah dari data desimal RC4 menjadi Karakter yang melalui proses *Base64*. Kode indeks *Base64* yang perlu diubah seperti simbol ‘+’ diubah menjadi ‘-‘ dan ‘/’ diubah menjadi ‘_’.

Data Hasil Pengujian

Pada tahap ini, data yang dikumpulkan dari hasil pengujian sistem dilakukan untuk mengamankan *QR Code* dari sistem E-Locker dengan menggunakan algoritma Rivest Cipher 4. Pengujian ini dilakukan dengan mengirimkan data dari Aplikasi Mobile ke ESP32-CAM dan ESP32-CAM ke *Server*. Pengujian sistem terdiri dari dua bagian utama yaitu Pengujian Proses Enkripsi dan Dekripsi Algoritma Rivest Cipher 4 untuk memastikan bahwa data enkripsi yang digunakan dalam verifikasi sesuai dengan proses dekripsi dan Pengujian Proses Waktu autentikasi Barcode *QR Code* ke ESP32-CAM untuk mengetahui kecepatan autentikasi *QR Code* pada ESP32-CAM.

Kesimpulan dan Saran

Pada tahap kesimpulan dan saran bagian akhir dari penelitian ini, yang dimana menyimpulkan hasil dari penelitian yang dilakukan. Kesimpulan diambil dengan menjawab pertanyaan dalam rumusan masalah yang didasarkan pada hasil pengujian sistem serta data yang diperoleh. Hasil penelitian juga menjadi dasar dalam memberikan saran untuk memperbaiki dan mengembangkan penelitian selanjutnya. Dengan demikian, hasil penelitian ini dapat memberikan posisi pada setiap proses keamanan.

HASIL DAN PEMBAHASAN

Hasil pengujian sistem dilakukan untuk mengamankan *QR Code* dari sistem E-Locker dengan menggunakan algoritma Rivest Cipher 4. Pengujian ini dilakukan dengan mengirimkan data dari Aplikasi Mobile ke ESP32-CAM dan ESP32-CAM ke *Server*. Pengujian sistem terdiri dari dua

bagian utama yaitu Pengujian Proses Enkripsi dan Dekripsi Algoritma Rivest Cipher 4 untuk memastikan bahwa data enkripsi yang digunakan dalam verifikasi sesuai dengan proses dekripsi dan Pengujian Proses Waktu autentikasi Barcode *QR Code* ke ESP32-CAM untuk mengetahui kecepatan autentikasi *QR Code* pada ESP32-CAM.

Pengujian ini dilakukan dengan pengguna yang sudah dibuatkan akun serta diberikan akses loker dari admin. Setelah pengguna melakukan login pada aplikasi mobile, pengguna dapat menggunakan barcode *QR Code* yang sudah dilakukan nya proses enkripsi dengan algortima RC4 apakah berhasil untuk melakukan perubahan menjadi data *ciphertext* dengan data asli dan dari hasil data *ciphertext* tersebut melakukan perubahan menjadi balok *QR Code*. Setelah pengguna melakukan autentikasi barcode *QR Code* ke ESP32-CAM akan mengirimkan data *ciphertext* ke *server* untuk melakukan proses dekripsi dan validasi data apakah sesuai dengan data asli dan berhasil. Hasil pengujian proses enkripsi dapat dilihat pada Tabel 8.

Tabel 8. Hasil Pengujian

NO.	Kunci	Data Asli (UUID)	Data Hasil (Enkripsi)
1.	*user*202509 29074047	1e67c366c062450383 d89e54cf469ece	17cAtfCqWkH8HySn2ZRwKMQw0QlV_Y-Lo7EgugOGCsM=
2.	*user*202509 29074047	69f593f7999e4182849 59d23b3d5271c	0OtQt6qqCkCmFivw2ZB4KcQ3jARV_IiMouRwuQjUWMU=
3.	*user*202509 29074047	de6603fd4c0e4ed1a65 79ffd36eac85e	grcAtKOqChOrTCLw2cQkKp01gAZV_tzb8-Fx7VnbXMM=
4.	*user*202509 29074047	fc6d55a233b34f7380c 69e0a0c9da9a6	gLEA5qasDUWsHHCm2cd3KMQz1gdV_Yre8LQt6FvaCJA=
5.	*user*202509 29074047	47f3fae82e5b4cb5a17 1323c7dbcf168	0uVQsfX4CU-tSif32cIiLp0yggBfqonc97N271zSX54=
…	…	…	…
39.	*user*202508 10125910	5f23492d58e3413a83b 1f76f934a1999	dCkqR-v88pp02cNjzXFsl4z_v3pkd9iaLP88FXuH5E8=
40.	*user*202508 10125910	595221b0e0a14104a2 430448ad6528d6	dHYtRu30os4k0cdhzXFvwtX-6XgydNrEdKg-QXiGuUA=

Hasil dari pengujian di atas kunci yang digunakan secara statis dan UUID yang digunakan secara dinamis dengan melakukan proses enkripsi RC4 di aplikasi mobile menghasilkan ciphertext yang berbeda dari UUID. Pengujian dilakukan dengan total sebanyak 40 data dari masing-masing 1 pengguna 10 data.

Pengujian proses waktu autentikasi *QR Code* ke ESP32-CAM, dilakukan dengan menghitung berapa lama ESP32-CAM membaca atau memindai *QR Code* dalam melakukan validasi sehingga membuka atau menutup Selonoid Door Lock. Proses perhitungan waktu autentikasi dilakukan dengan menggunakan *Stopwatch headphone* dengan dimulai saat *QR Code* di autentikasi Ke ESP32-CAM. Setelah itu menunggu reaksi dari ESP32-CAM menvalidasi dari data apakah sesuai dengan yang sudah ada si *server*. Jika sesuai maka ESP32-CAM akan menerima Kembali data dan mengirimkan sinyal notifikasi bunyi pada buzzer sebanyak 1 kali. Hasil pengujian *waktu response time* dapat dilihat pada Tabel 9.

Tabel 9. Hasil Pengujian *Response Time*

NO.	Data Asli (UUID)	Data Hasil Waktu (Detik)
1.	1e67c366c062450383d89e54cf469ece	11.28
2.	69f593f7999e418284959d23b3d5271c	6.59
3.	de6603fd4c0e4ed1a6579ffd36eac85e	4.11
4.	fc6d55a233b34f7380c69e0a0c9da9a6	7.54
5.	3573c1b175fa489e92b78942610d5490	2,83
…	…	…
40	595221b0e0a14104a2430448ad6528d6	4,04
	Rata-rata	5.42

Hasil dari pengujian diatas yang dimana menggunakan total 40 data dengan 1 pengguna melakukan 10 data percobaan dalam melakukan autentikasi barcode *QR Code* ke ESP32-CAM. Dari 40 data tersebut ESP32-CAM dapat melakukan autentikasi barcode *QR Code* dengan waktu tercepat 2.83 detik, waktu autentikasi lama 12.41 detik dan waktu rata-rata yang didapatkan dari keseluruhan 5,42 detik.

Pembahasan

Berdasarkan hasil dari pengujian yang telah dilakukan pada sistem E-Locker dengan menerapkan metode kriptografi algortima Rivest Cipher 4, dapat disimpulkan bahwa sistem ini mampu memenuhi dalam implementasi keamanan dengan memposisikan letak bagian proses enkripsi dan dekripsi pada saat pengiriman data. Data yang dienkripsi yaitu, UUID yang akan digunakan menjadi data *Ciphertext* untuk barcode *QR Code*. Pengujian dilakukan Proses Enkripsi dan Dekripsi Algoritma Rivest Cipher 4 dan Proses Waktu Auntentikasi Barcode *QR Code* ke ESP32-CAM. Data hasil dari pengujian yang digunakan sebanyak 40 data dari setiap pengguna melakukan akses sebanyak 10 kali autentikasi. Pada pengujian keamanan data UUID, setelah *sniffing* berhasil mendapatkan data asli yang dikonversi ke barcode *QR Code* dengan menggunakan *website* Generation *QR Code*. Menghasilkan autentikasi gagal dengan menggunakan data asli UUID saat autentikasi ke ESP32-CAM. Waktu yang didapatkan dalam melakukan autentikasi Barcode *QR Code* dari aplikasi mobile ke ESP32-CAM menggunakan algoritma Rivest Cipher 4 dengan waktu terendah 2,83 detik sementara ditempuh waktu terlama 12,41 detik dan rata-rata yang didapatkan 5,42 detik.

Secara keseluruhan, pengujian membuktikan bahwa metode Rivest Cipher 4 dapat diterapkan dalam sistem E-Locker. Metode ini mampu menjaga kerahasian data dimana memposisikan bagian proses enkripsi dan dekripsi. Dengan demikian, sistem E-Locker yang telah diterapkan dengan Algortima Rivest Cipher 4 dapat mengabungkan aspek keamanan dan efesiensi.

SIMPULAN

Hasil Pengujian berhasil dengan menggunakan 40 data dari setiap pengguna melakukan 10 kali autentikasi *QR Code*, data UUID yang digunakan akan selalu berubah saat melakukan autentikasi *QR Code* dan Kunci yang digunakan berasal dari *role* pengguna dan waktu kapan dibuat yang digabungkan sehingga menjadi kunci. Proses yang diterapkan dari Algortima Rivest Cipher 4, proses enkripsi yang dilakukan di Aplikasi Android yang dimana mengamankan data asli UUID menjadi data ciphertext dan Proses dekripsi yang dimana membaca dari data *ciphertext* ke *Plaintext* yang dilakukan di *server* serta validasi *QR Code* untuk notifikasi jika berhasil atau gagal dari Buzzer. Hasil pengujian keamanan pada data asli UUID nya yang dimana jika data asli UUID dapat diketahui *sniffing* sehingga data asli dikonversi ke *QR Code* tersebut digunakan untuk mengakses loker maka tidak berhasil dikarenakan untuk mengakses loker menggunakan data *ciphertext* yang sudah menjadi *QR Code* di aplikasi mobile sehingga data *ciphertext* tersebut tidak tersimpan di *server*.Waktu dalam melakukan Autentikasi *QR Code* ke ESP32-CAM dengan melakukan pengujian sebanyak 40 data, dengan waktu yang dibutuhkan tercepat 2,83 detik untuk mendeteksi *QR Code* saat membuka Solenoid Door Lock, waktu terlama yang dibutuhkan 12,41 detik dan rata-rata yang didapatkan 5,42 detik.

DAFTAR PUSTAKA

[1] I. Komang, “Rancang Bangun Sistem Pengunci Loker Otomatis Dengan Kendali Akses Menggunakan Rfid Dan Sim 800L,” *J. Ilm. Mhs. Kendali dan List.*, vol. 1, no. 1, pp. 33–41, 2020, doi: 10.33365/jimel.v1i1.187.

[2] R. Putra, A. Uperiati, T. Suhendra, and D. Amalia Purnamsari, “Implementasi Sistem Keamanan Loker Berbasis Wireless Sensor Network di Perpustakaan Pusat UMRAH,” vol. 12, no. 01, p. 2023, 2023.

[3] N. Rubiati and S. W. Harahap, "Aplikasi Absensi Siswa Menggunakan *QR Code* Dengan Bahasa Pemrograman Php Di Smkit Zunurain Aqila Zahra Di Pelintung," *I N F O R M a T I K a*, vol. 11, no. 1, p. 62, 2019, doi: 10.36723/juri.v11i1.156.

[4] M. R. Abdahu, U. Ristian, and H. Hasfani, "Implementasi Smart Helmet Cabinet pada Penyimpanan Helm Berbasis Mobile *QR Code*," *J. Inform. J. Pengemb. IT*, vol. 9, no. 1, pp. 78–85, 2024, doi: 10.30591/jpit.v9i1.6593.

[5] D. R. Saragi, J. M. Gultom, J. A. Tampubolon, and I. Gunawan, "Pengamanan Data File Teks (Word) Menggunakan Algoritma RC4," *J. Sist. Komput. dan Inform.*, vol. 1, no. 2, p. 114, 2020, doi: 10.30865/json.v1i2.1745.

[6] M. Alfan, S. Sutardi, and B. Pramono, "Enkripsi Data *QR Code* Menggunakan Metode Rc4 Pada Aplikasi Presensi Jurusan Teknik Informatika Universitas Halu Oleo," *semanTIK*, vol. 7, no. 2, p. 191, 2021, doi: 10.55679/semantik.v7i2.8927.

[7] D. Mukminin, A. Ameliana, A. Dwikanthi, and L. U. Zogara, "Aplikasi Kehadiran Siswa Berbasis Android Menggunakan *QR Code* Di Bimbel Excellent Institute," *J. Teknoinfo*, vol. 16, no. 1, p. 66, 2022, doi: 10.33365/jti.v16i1.1362.

[8] D. Setiawan, H. Jaya, S. Nurarif, T. Syahputra, M. Syahril Syafnur, and S. Triguna Dharma, "Arduino Ide, Rangkaian Penurun Tegangan," *J. Sci. Soc. Res.*, vol. 4307, no. 1, pp. 159–164, 2022, [Online]. Available: http://jurnal.goretanpena.com/index.php/JSSR

[9] F. Nadziroh, F. Syafira, and S. Nooriansyah, "Alat Deteksi Intensitas Cahaya Berbasis Arduino Uno," *Indones. J. Intellect. Publ.*, vol. 1, no. 3, pp. 142–149, 2021, doi: 10.51577/ijipublication.v1i3.92.

[10] R. Juliansyah, E. Fitriani, N. Paramita, and ..., "Rancang Bangun Sistem Kontrol Motor Feeder dan Monitoring Pakan Ikan Nila Berbasis Smart Relay Zelio," *J. Pendidik. …*, vol. 8, pp. 11157–11167, 2024, [Online]. Available: https://www.jptam.org/index.php/jptam/article/view/14054%0Ahttps://www.jptam.org/index.php/jptam/article/download/14054/10820

[11] Zeluyvenca Avista and Oldy Fahlovi, "Rancang Bangun Smart Door Access Berbasis Fingerprint untuk Keamanan Ruang Laboratorium," *Venus J. Publ. Rumpun Ilmu Tek.*, vol. 2, no. 1, pp. 01–13, 2024, doi: 10.61132/venus.v2i1.73.

[12] M. Azhari, D. I. Mulyana, F. J. Perwitosari, and F. Ali, "Implementasi Pengamanan Data pada Dokumen Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES)," *J. Pendidik. Sains dan Komput.*, vol. 2, no. 01, pp. 163–171, 2022, doi: 10.47709/jpsk.v2i01.1390.

[13] R. B. Nuerita Maharani, M. I. P. Nasution, and T. Triase, "Sistem Informasi Payroll Pegawai dengan Absensi *QR Code*," *J. Inform. dan Teknol. Pendidik.*, vol. 1, no. 1, pp. 23–35, 2021, doi: 10.25008/jitp.v1i1.9.

[14] S. K. Dirjen, P. Riset, D. Pengembangan, R. Dikti, M. R. Royani, and A. Wibowo, "Terakreditasi SINTA Peringkat 2 Implementasi Web Service pada Perusahaan Logistik menggunakan JSON Web Token dan Algoritma Kriptografi RC4," Masa Berlaku Mulai, vol. 1, no. 3, pp. 591–600, 2017.

[15] M. Efendi, V. Sihombing, and S. Parulian, "Implementation and Use of Base64 Algorithm in Video File Security," Sinkron, vol. 7, no. 1, pp. 243–247, 2022, doi: 10.33395/sinkron.v7i1.11256.