Terjemahan dari 10.22146/jnteti.v13i1.4395

Implementasi Sistem Keamanan Presensi Berbasis Kode QR Menggunakan Algoritma RSA dan *Hash*

Arif Indra Irawan[1], Iman Hedi Santoso[1], Istikmal[1], Maya Rahayu[2]

[1] Program Studi Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom, Bandung, Indonesia
[2] Program Studi D-3 Teknik Telekomunikasi, Jurusan Teknik Elektro, Politeknik Negeri Bandung, Bandung, Indonesia

[Diserahkan: 3 Agustus 2022, Direvisi: 11 Agustus 2023, Diterima: 8 Desember 2023]
Penulis Korespondensi: Arif Indra Irawan (email: arifirawan@telkomuniversity.ac.id)

INTISARI — Aplikasi presensi berbasis kode *quick response* (QR) berkontribusi dalam mengurangi penggunaan kertas dan dapat menekan kesalahan masukan presensi. Akan tetapi, implementasi aplikasi presensi siswa menggunakan kode QR menunjukkan kerentanan dalam proses penggunaannya di sebuah sekolah di Bandung. Siswa dapat membuat presensi palsu untuk diri sendiri atau siswa lain yang tidak hadir ke kelas. Serangan seperti ini dikenal sebagai pembangkitan kode QR palsu (*fake QR code generation*). Penelitian ini mengusulkan sistem autentikasi keamanan menggunakan algoritma enkripsi Rivest–Shamir–Adleman (RSA) dan algoritma *secure hash algorithm* 1 (SHA-1) untuk mengamankan aplikasi presensi berbasis kode QR dari serangan pembangkitan kode QR palsu. Algoritma RSA digunakan untuk mengenkripsi data kode QR untuk menjaga privasi, sedangkan algoritma SHA-1 digunakan memastikan integritas data. Dengan menggunakan metode ini, proses *mutual authentication* antara data kode QR yang dihasilkan oleh siswa dan aplikasi pembacaan presensi oleh guru dapat terjalin. Hasil yang diperoleh dari serangkaian pengujian menyatakan bahwa sistem keamanan pada aplikasi presensi siswa yang telah diterapkan di Madrasah Aliyah (MA) Al Mukhlishin dapat mendeteksi dan mencegah terjadinya serangan pembangkitan kode QR palsu. Kemudian, pengujian dilakukan dengan mengubah pengaruh panjang kunci pada RSA 1.024 bit dan RSA 2.048 bit. Hasilnya menunjukkan bahwa pada RSA 1.024 bit, digunakan energi sebesar 0,14 J dan waktu 1,66 s, lebih efisien dibandingkan dengan RSA 2.048 bit, dengan konsumsi energi 0,19 J dan waktu 2,09 s. Menariknya, jika tingkat keamanan yang lebih tinggi diperlukan, panjang kunci harus ditingkatkan dengan mengorbankan beberapa efisiensi energi dan waktu.

KATA KUNCI — Sistem Autentikasi, Daftar Hadir Siswa, Kode QR, RSA, *Hash*.

I. PENDAHULUAN

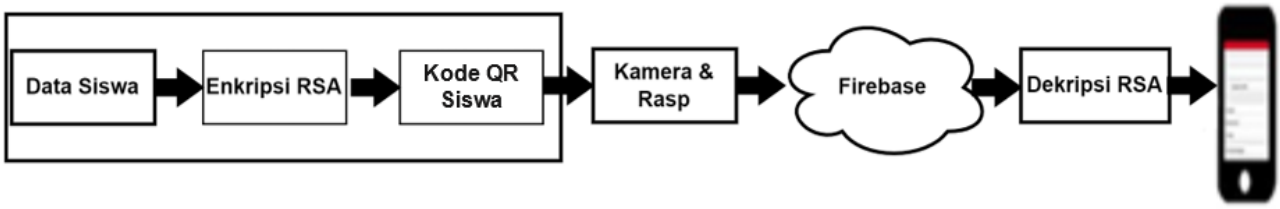
Kode *quick response* (QR) saat ini telah banyak digunakan dalam berbagai aplikasi, seperti promosi produk, identitas pembayaran listrik, pembayaran kredit telepon seluler, layanan kesehatan [1], dan robot bergerak, bersamaan dengan perkembangan teknologi ponsel pintar (*smartphone*) [2]. Lebih jauh lagi, penggunaan kode QR juga dapat melengkapi aplikasi pemantauan dan otomatisasi lainnya, seperti sistem pintar [3] dan Bluetooth [4]. Hal ini terjadi karena proses *decoding* semua versi kode QR dengan menggunakan mata manusia akan sangat sulit dilakukan karena pesan tersebut dienkripsi menjadi bit-bit yang kemudian membentuk sebuah larik persegi [5].

Kemudian, kode QR saat ini juga sudah bisa digunakan untuk melengkapi data kehadiran siswa di sekolah. Berbagai macam cara dapat dilakukan untuk mencatat kehadiran siswa, misalnya dengan cara manual menggunakan kertas, menggunakan teknologi *radio frequency identification* (RFID) [6], atau menggunakan kode QR yang memanfaatkan *smartphone*. Namun, di antara cara-cara presensi di atas, hanya kode QR yang saat ini memiliki potensi untuk diaplikasikan dalam sistem presensi siswa, seiring dengan perkembangan *smartphone* karena *smartphone* dapat dimiliki oleh siapa pun dan mudah digunakan. Selain itu, penggunaan kode QR dapat mengurangi penggunaan kertas sebagai media presensi siswa yang dianggap tidak efisien dan mahal [7]. Metode presensi siswa tanpa kertas ini akan berdampak positif pada masalah lingkungan dan meningkatkan kualitas pendidikan yang didukung oleh efektivitas sistem administrasi.

Di sisi lain, sistem presensi siswa berbasis kode QR memiliki kerentanan seperti *phishing* kode QR [8], pembuat QR palsu, *malicious QR code*, dan peretasan data [9]. Jenis eksploitasi ini memungkinkan siswa untuk melakukan kecurangan dalam sistem presensi. Kode QR merupakan sebuah kode pengenalan objek atau produk yang pertama kali dikembangkan oleh perusahaan Jepang, Wave Denso Company [10]. Kode QR memiliki bentuk kode matriks atau kode batang dua dimensi [11]. Teknologi kode QR telah banyak digunakan untuk mengidentifikasi dan mengenali berbagai produk, termasuk sistem pembayaran daring dan digital [10], [11].

Salah satu contoh eksploitasi kode QR yang teridentifikasi adalah siswa dapat menyalin format pesan dalam kode QR menggunakan alat pemindai, lalu menciptakan ulang pesan kode QR, atau yang biasa disebut pembangkitan kode QR palsu (*fake QR code generation)*, untuk menipu sistem presensi siswa. Dengan cara ini, siswa yang sebenarnya tidak menghadiri kelas dapat melakukan presensi dengan menitipkan data kode QR ke siswa yang hadir. Oleh karena itu, sistem keamanan yang baik diperlukan untuk meningkatkan keamanan sistem presensi berbasis kode QR. Maka, penelitian ini mengusulkan sistem autentikasi berbasis kode QR menggunakan algoritma Rivest–Shamir–Adleman (RSA) dan *hash*. Sistem autentikasi ini telah dikembangkan dan diuji coba di sebuah sekolah di Bandung. Hasil pengujian menunjukkan kemampuan sistem dalam mencegah *spoofing* dan pembangkitan kode QR palsu.

Berbagai penelitian tentang kode QR telah dilakukan, seperti penelitian untuk meningkatkan kinerja visual QR dengan menggunakan metode *halftone* dan algoritma koreksi kesalahan sistematis Berlekamp Reed-Solomon [12]. Banyak penelitian yang telah membahas peningkatan keamanan sistem



Gambar 1. Model sistem presensi siswa.

kode QR ini dengan menggabungkan berbagai teknik, seperti menambahkan lapisan kode QR [13], menggunakan *proxy re-encryption* [14], kriptografi [15]-[17], *watermark* [18], *blockchain* [19], dan steganografi [20].

Dalam sebuah penelitian, diperkenalkan tipe baru kode QR yang disebut sebagai kode QR dua level (*two-level QR code*, 2LQR), yang terdiri atas level publik dan level privat [21]. Level publik berfungsi mirip dengan level penyimpanan kode QR standar dan dapat dibaca oleh aplikasi kode QR tradisional. Di sisi lain, level privat membutuhkan aplikasi khusus dan informasi masukan khusus. Kode 2LQR ini dapat digunakan untuk berbagi pesan pribadi dan autentikasi. Pada penelitian lain, diusulkan kode QR dinamis untuk sistem pembayaran yang mendukung algoritma kriptografi SM2, SM3, dan SM4 [22]. Perbandingan dilakukan antara algoritma-algoritma ini dengan algoritma yang lain, seperti *advanced encryption standard* (AES) dan RSA, dengan menggunakan keacakan dari *ciphertext* dan waktu komputasi sebagai parameter untuk mengukur kinerja keamanan.

Sementara itu, berbeda dengan penelitian-penelitian sebelumnya, protokol autentikasi pada penelitian ini diimplementasikan dengan menggunakan algoritma enkripsi yang kuat, kemudian di sisi server, hasil enkripsi (*ciphertext*) tersebut disimpan menggunakan salinan *hash* dari kata sandi tersebut. Selanjutnya, algoritma enkripsi RSA yang digunakan dalam penelitian ini merupakan sistem kriptografi kunci publik, yang proses enksripsinya hanya memerlukan satu pasangan kunci yang digunakan secara bersamaan [23]. Keunggulan dari algoritma ini terletak pada proses eksponensial dan faktorisasi angka menjadi dua bilangan prima, yang sampai saat ini memerlukan waktu yang lama untuk difaktorkan. Algoritma ini dinamai berdasarkan para penemu, yaitu Ron Rivest, Adi Shamir, dan Leonard Adleman (Rivest-Shamir-Adleman), dan dipublikasikan pada tahun 1977 di MIT sebagai tanggapan terhadap tantangan yang diajukan oleh algoritma pertukaran kunci Diffie-Hellman. Skema RSA mengadopsi skema blok *cipher*, yaitu sebelum enkripsi dilakukan, *plaintext* yang ada dibagi menjadi blok-blok dengan panjang yang sama. *Plaintext* dan *ciphertext* adalah bilangan bulat antara 1 hingga *n*, dengan *n* biasanya berukuran 1.024 bit dan panjang blok kurang dari atau sama dengan log(*n*) + 1 dengan basis 2 [23].

Algoritma *hash* adalah algoritma enkripsi untuk mengubah teks menjadi serangkaian karakter acak [24]. Jumlah karakter dalam hasil *hash* selalu sama. *Hash* merupakan enkripsi satu arah, sehingga pesan yang telah di-*hash* tidak dapat dikembalikan menjadi teks aslinya. *Secure hash algorithm* 1 (SHA-1) merupakan salah satu dari banyak algoritma *hashing* yang digunakan secara umum untuk memastikan integritas data. Dalam penelitian ini, SHA-1 diimplementasikan menggunakan *library* hashlib. Pesan hasil *hash* SHA-1 yang digunakan dalam penelitian ini memiliki lebar data sebesar 20 byte, yang ditampilkan sebagai angka heksadesimal 40 digit. Diyakini bahwa ini adalah algoritma terbaik untuk meningkatkan sistem keamanan.

II. METODOLOGI

Desain sistem diilustrasikan melalui diagram blok dan diagram alir yang menjelaskan cara kerja sistem. Metode dari sistem ini dibuat dengan merancang sistem keamanan mengikuti diagram alir yang telah dibuat sebelumnya, kemudian menganalisis kembali kevalidan desain yang dibuat. Setelah itu, sistem diimplementasikan dengan menerapkan konsep yang telah dibuat ke dalam server *cloud* dan aplikasi Android. Lalu, keberhasilan implementasi sistem diperiksa. Keberhasilan ditandai dengan tidak adanya *bug* atau kesalahan.

A. SISTEM DESAIN

Perancangan sistem pada proses presensi siswa menggunakan kode QR ini memiliki beberapa tahapan yang harus diselesaikan, seperti ditunjukkan pada Gambar 1. Proses pengambilan data kehadiran siswa dengan menggunakan kode QR dimulai dengan guru dan siswa mengunduh aplikasi Android yang menggunakan sistem keamanan yang ditingkatkan. Selanjutnya, jika baru pertama kali menggunakannya, pengguna (baik siswa maupun guru) harus melakukan registrasi terlebih dahulu. Saat kelas dimulai, siswa memindai kode QR yang dihasilkan oleh *smartphone* masing-masing dengan menggunakan modul kamera yang terhubung dan menyimpannya di basis data server Firebase. Kode QR siswa dihasilkan dengan mengenkripsi data siswa menggunakan algoritma enkripsi RSA. Pembacaan kode QR siswa dengan modul kamera diatur dan diproses oleh Raspberry Pi. Selanjutnya, guru dapat melihat daftar kehadiran siswa di *smartphone* guru setelah *login* ke aplikasi guru. Seluruh pesan telah melalui autentikasi dari Raspberry Pi.

Setelah langkah-langkah awal terpenuhi, ditentukan alur penggunaan data dari langkah awal sampai menjadi sebuah catatan kehadiran siswa yang dapat diakses oleh guru dan staf administrasi di sekolah. Proses ini harus dilakukan secara berurutan atau *sequential*. Alur proses presensi menggunakan kode QR ditunjukkan pada Gambar 2.

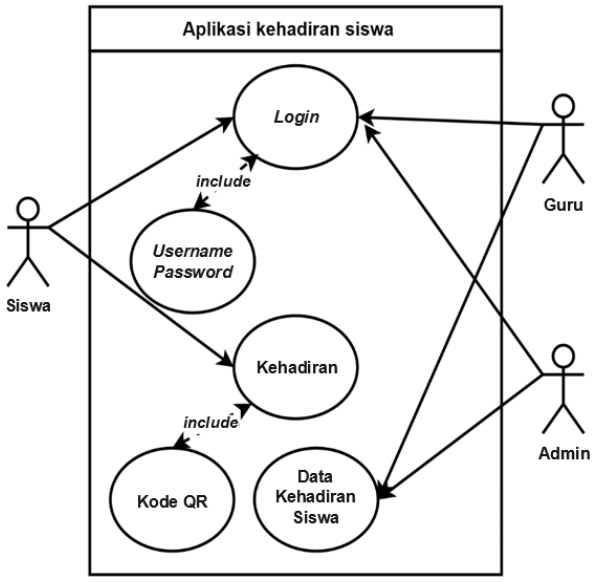
Gambar 2 menunjukkan diagram *use case* sistem secara umum. Sistem ini memiliki tiga entitas, yaitu siswa, guru, dan admin. Siswa harus masuk ke aplikasi sebelum dapat menghasilkan kode QR. Raspberry Pi kemudian membaca kode QR yang dihasilkan, memprosesnya, dan mengirimkannya ke basis data sebagai daftar kehadiran siswa. Guru dapat melihat status siswa dari daftar kehadiran siswa di basis data, sedangkan admin bertanggung jawab untuk mengelola basis data tersebut.

B. DESAIN APLIKASI PRESENSI SISWA

Dalam aplikasi presensi siswa, terdapat beberapa fitur, antara lain sebagai berikut.

1) HALAMAN PENDAFTARAN

Pada halaman pendaftaran terdapat beberapa kotak masukan yang harus diisi oleh siswa Madrasah Aliyah (MA) Al Mukhlishin Bandung, seperti nama, NISN, nama pengguna, *email*, dan kata sandi. Setelah mengisi data, siswa dapat



Gambar 2. Diagram *use case* sistem secara umum.

menekan tombol kirim untuk mentransfer data ke server basis data.

2) HALAMAN *LOGIN* SISWA

Halaman *login* siswa berisi dua kotak masukan, yaitu nama pengguna dan kata sandi. Ketika siswa menekan tombol *login*, aplikasi akan melakukan autentikasi dengan data yang disimpan dalam basis data ketika pendaftaran.

3) APLIKASI PROFIL SISWA

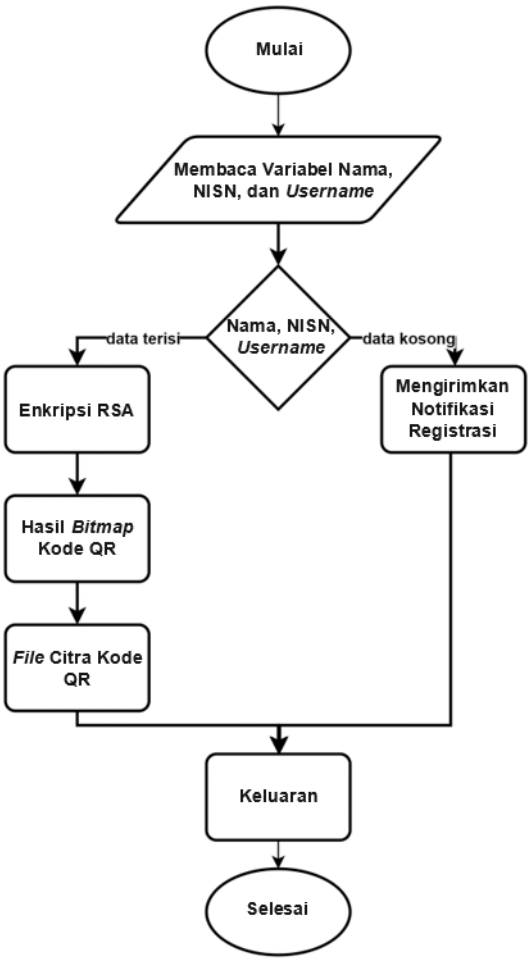
Halaman profil siswa berisi tiga kolom, yaitu nama, NISN, dan gambar kode QR, yang dihasilkan secara dinamis berdasarkan data yang didaftarkan oleh siswa. Gambar kode QR yang dihasilkan pada halaman ini menentukan kehadiran siswa.

4) NOTIFIKASI KEHADIRAN

Fitur notifikasi terdapat pada halaman yang sama dengan aplikasi profil siswa. Notifikasi profil siswa muncul jika siswa berhasil hadir.

Semua fitur yang ada pada aplikasi presensi siswa tersebut memiliki alur proses spesifik yang secara teknis dapat digambarkan seperti pada Gambar 3. Diagram alir tersebut menjelaskan bahwa siswa harus membuat akun dengan memasukkan nama lengkap, NISN, dan nama pengguna. Ketika siswa mencatatkan kehadiran, ditambahkan penanda waktu (*timestamp*) pada data, kemudian data dienkripsi dengan teknik RSA dan menghasilkan sebuah *ciphertext*. *Ciphertext* ini akan didekode menjadi sebuah kode QR menggunakan pustaka ZXing dan hasilnya berupa gambar *bitmap*. Gambar *bitmap* hasil pengodean ini akan mencatat kehadiran siswa dan dapat dipindai dengan kamera Raspberry Pi. Jika siswa tidak mengisi nama lengkap, NISN, dan nama pengguna, program akan memberi notifikasi kepada siswa untuk mendaftar.

Dengan sistem autentikasi ini, format data kehadiran tidak dapat dibaca jika siswa menggunakan aplikasi pihak ketiga untuk membaca kode QR. Jika seorang siswa ingin membuat kode QR palsu, siswa hanya dapat membaca dan menduplikasi data kode QR yang sebelumnya telah dienkripsi. Terlebih lagi, dengan adanya penanda waktu, tindakan ini dapat diidentifikasi, sehingga serangan kode QR palsu dapat terdeteksi dan diblokir. Untuk merusak autentikasi, penyerang perlu membobol RSA 1.024 atau 2.048 bit, yang hingga saat ini belum berhasil dipecahkan. RSA terbesar yang pernah berhasil



Gambar 3. Diagram alir aplikasi kehadiran siswa.

dipecahkan adalah RSA 768-bit yang dilakukan oleh Paul Zimmermann dkk. pada tanggal 12 Desember 2009 dan memerlukan waktu selama dua tahun [25].

C. DESAIN PROGRAM RASPBERRY PI

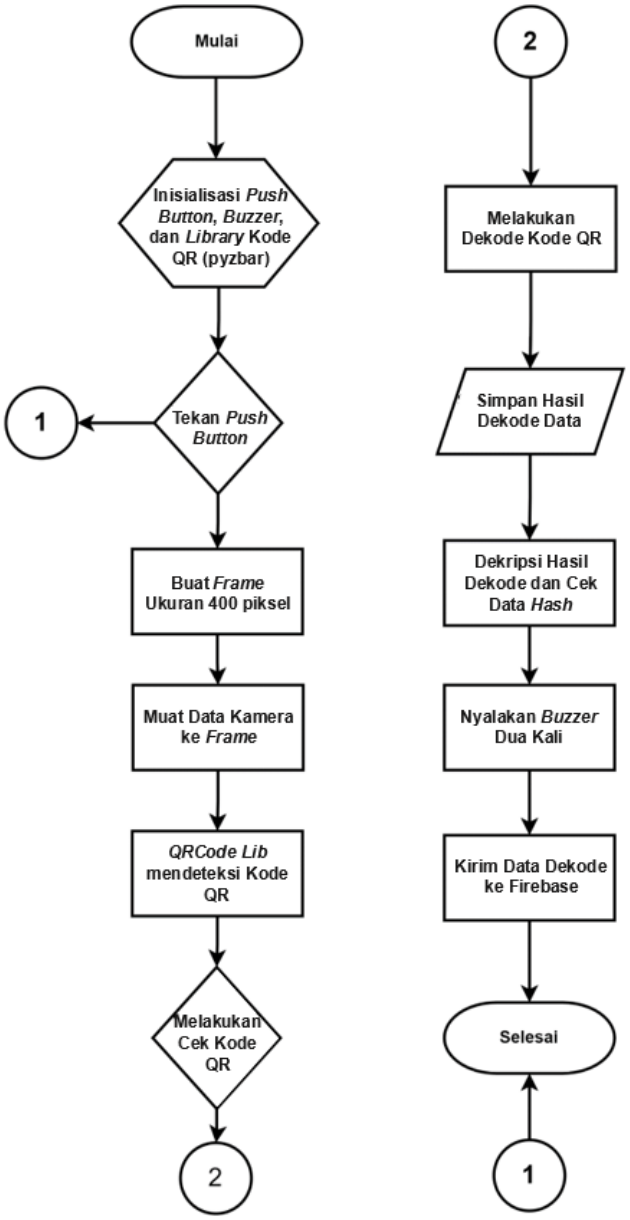
Pada penelitian ini, Raspberry Pi berfungsi sebagai pengontrol untuk membaca data kehadiran siswa dari kode QR yang dihasilkan oleh aplikasi siswa. Siswa perlu memindai kode QR dari aplikasi ke kamera web pada Raspberry Pi. Raspberry Pi dilengkapi dengan *buzzer* untuk memeriksa kode QR siswa telah berhasil dibaca atau belum. Komponen terakhir adalah *push button* yang digunakan untuk mereset program Raspberry Pi jika terjadi kesalahan atau *bug*. Dua komponen di atas terhubung ke *pin* digital pada Raspberry Pi, seperti yang ditunjukkan pada Gambar 4.

Gambar 4 menunjukkan diagram alir program Raspberry Pi. Program melakukan inisialiasi komponen yang terhubung ke Raspberry Pi, seperti *push button* dan *buzzer*. Selain itu, pyzbar juga dimasukkan dalam perangkat lunak untuk memproses kode QR. Aplikasi akan membangun bingkai (*frame*) berukuran 400 piksel dan menggunakan data kamera ini untuk menentukan ada tidaknya kode QR. Jika *lybrary* pyzbar mendeteksi kode QR, program akan menampilkan sebuah kotak yang mengelilingi kode QR dan mengaktifkan *buzzer* dua kali, menandakan bahwa data kode QR telah terbaca. Pada proses terakhir, data yang dienkripsi ini akan ditransmisikan ke Firebase.

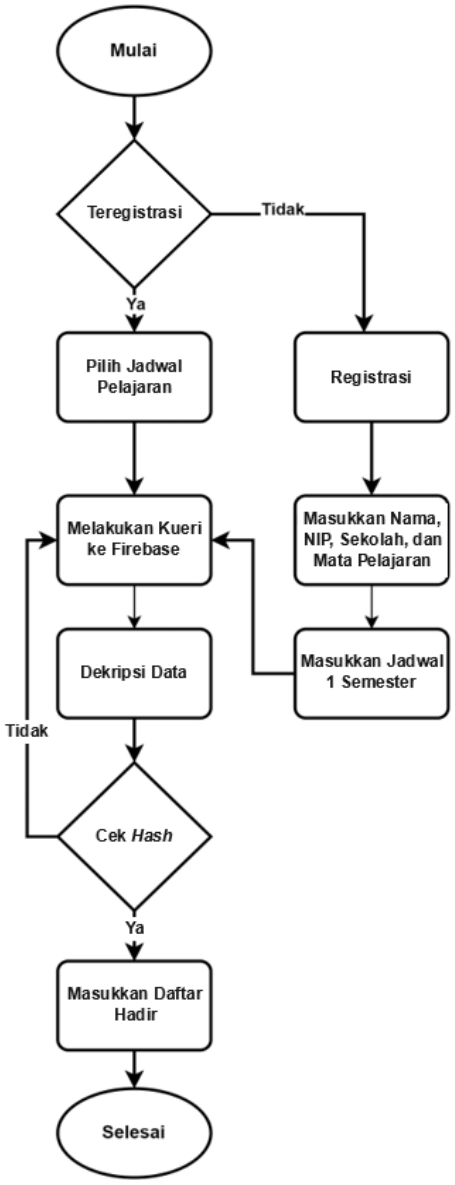
D. DESAIN APLIKASI PRESENSI GURU

Aplikasi guru menampilkan beberapa fitur seperti berikut.

p-ISSN 2301–4156 | e-ISSN 2460–5719



Gambar 4. Diagram alir program Raspberry Pi.



Gambar 5. Diagram alir aplikasi kehadiran guru.

1) REGISTRASI

Halaman registrasi berisi beberapa kotak masukan yang dapat diisi oleh guru di MA Al Mukhlishin Bandung, seperti nama pengguna, nama lengkap, NIP, *email*, dan kata sandi untuk masuk. Setelah guru menekan tombol registrasi, data akan ditransfer ke server basis data.

2) HALAMAN *LOGIN* GURU

Halaman *login* berisi beberapa kotak masukan seperti nama pengguna dan kata sandi. Ketika guru menekan tombol *login*, program akan melakukan autentikasi menggunakan informasi yang dimasukkan pada saat pendaftaran.

3) APLIKASI PROFIL GURU

Halaman profil berisi data kehadiran siswa yang dihasilkan pada jam pelajaran tertentu. Alur desain aplikasi presensi guru ini diperlihatkan pada Gambar 5. Tampak pada gambar bahwa setelah membuat akun, guru dapat membuat laporan kehadiran dan jadwal pelajaran. Data laporan kehadiran diambil dari kueri yang dibuat dari Firebase. Setelah aplikasi guru menerima data tersebut, data akan didekripsi menggunakan kunci privat yang disimpan dalam aplikasi. Jika data siswa tidak berhasil didekripsi atau tidak sesuai dengan jadwal yang ditentukan oleh guru, data tersebut akan dianggap tidak valid dan siswa tidak akan terdaftar dalam laporan kehadiran.

Proses verifikasi dilakukan pada data siswa dalam aplikasi guru dengan memanfaatkan kriptografi asimetris dan penanda waktu. Beberapa skenario serangan eksperimental telah dibuat untuk menguji kinerja sistem autentikasi, seperti menduplikasi data kode QR dan mengonstruksi kode QR siswa dengan jadwal pelajaran. Selama implementasi sistem, ditemukan bahwa aplikasi tidak dapat membaca beberapa data meskipun formatnya benar. Ditemukan bahwa fitur enkripsi dalam kode QR dapat mengurangi jarak baca kode QR. Masalah ini dapat diatasi dengan memindai ulang kode QR ke pemindai selama kelas berlangsung.

E. SISTEM KEAMANAN

Sistem keamanan yang digunakan dalam penelitian ini adalah algoritma RSA untuk menjamin kerahasiaan data dan SHA-1 untuk menjamin integritas data. Pesan yang disematkan dalam kode QR diimplementasikan dalam aplikasi di *smartphone* siswa, yang dijelaskan sebagai berikut.

1. Pilih *N*, yaitu ukuran gambar kode QR dengan lebar = 350 dan tinggi = 350.

2. Pilih pesan apa pun yang dihasilkan dari tiga bidang: nama pengguna, nama siswa, dan kata sandi yang telah di-*hash*.
3. Buat muatan (*payload*) *ms* dan enkripsi muatan tersebut dengan algoritma RSA, lalu dapatkan *ciphertext S* menggunakan kunci publik dari server *Ku*.
4. Kemudian, hasilkan kode QR menggunakan *N*, *S* + *Tn*, dengan *Tn* adalah penanda waktu dari setiap pelajaran.

Proses ekstraksi kode QR diimplementasikan dalam Raspberry Pi dan aplikasi seluler guru yang dijelaskan sebagai berikut.

1. Baca kode QR yang diterima untuk mendapatkan *S*+*Tn* dengan menentukan dimensi kode QR *N*.
2. Ekstrak *S* dan *Tn*, lalu dekripsi *S* menggunakan kunci privat dari server *Kp*.
3. Autentikasi setiap kolom data dalam basis data.
4. Jika data berhasil diautentikasi, kirimkan *S* ke basis data.
5. Aplikasi pada *smartphone* guru dapat mengambil *S* dari basis data, mendekripsinya menggunakan *Kp*, lalu menampilkannya pada aplikasi guru.

F. DESAIN PERANGKAT KERAS

Penelitian ini menggunakan perangkat keras berupa Raspberry Pi untuk mendekode data kode QR dari kamera, mendekripsi data, dan melakukan proses autentikasi kode QR. Data autentikasi yang valid kemudian dikirimkan ke basis data. Raspberry Pi juga dilengkapi dengan beberapa fitur lain, seperti penanda untuk memastikan kode QR dapat dibaca/diautentikasi atau tidak dengan menggunakan *buzzer*; dan tombol reset untuk mereset program pada Raspberry Pi jika terjadi masalah.

G. SKENARIO PENGUJIAN

Pengujian kinerja dilakukan pada perangkat Android dan perangkat Raspberry Pi untuk menilai kinerja aplikasi setelah penerapan metode autentikasi ini.

1) PENGUJIAN PROGRAM ANDROID

Pengujian aplikasi Android dibagi menjadi dua jenis, yaitu pengujian waktu eksekusi program dan konsumsi daya menggunakan perangkat lunak simulasi di Android Studio. Pengujian dilakukan sebanyak 30 kali secara keseluruhan. Menu *login* dan laporan kehadiran adalah dua elemen program yang diukur dengan waktu eksekusi. Waktu eksekusi setiap komponen program dijumlahkan dalam pengujian eksekusi program. Sementara itu, pengujian konsumsi daya dilakukan dengan kode QR dari siswa yang sebenarnya.

2) PENGUJIAN PROGRAM RASPBERRY PI

Pengujian dilakukan pada Raspberry Pi untuk mengukur kinerja proses pembacaan kode QR dan proses enkripsi. Parameter yang digunakan untuk mengukur kinerja Raspberry Pi adalah waktu eksekusi dan penggunaan memori. Aplikasi pihak ketiga, modul timeit, digunakan untuk mengukur waktu eksekusi program, sementara aplikasi top digunakan untuk mengukur penggunaan memori.

III. HASIL DAN DISKUSI

Tujuan penelitian ini adalah mengembangkan sistem presensi kode QR yang aman bagi setiap siswa di MA Al Mukhlishin Bandung untuk setiap pelajaran yang dihadiri. Sistem ini mirip dengan penelitian terdahulu, yang menciptakan kode QR dinamis untuk mencegah serangan pada kode QR [22]. Perbedaannya terletak pada perangkat keras dan algoritma yang digunakan. Penelitian sebelumnya menggunakan perangkat keras yang lebih mahal dan algoritma enkripsi yang terstandar untuk keuangan [22]. Sementara itu, dalam penelitian ini perangkat diuji berdasarkan parameter kinerja komputasi, waktu optimal, dan jarak yang dapat bekerja dengan sistem. Tampilan aplikasi ditunjukkan pada Gambar 6.

Gambar 6. Implementasi sistem.

Ketika pelajaran dimulai, siswa memindai gambar kode QR yang aman yang dihasilkan oleh *smartphone* setiap siswa. Kode QR unik ditangkap menggunakan modul kamera dan prosesor Raspberry Pi akan mengartikan data tersebut. Fungsi-fungsi ini telah diimplementasikan dalam aplikasi Android siswa, aplikasi Android guru, basis data, dan Raspberry Pi, seperti diperlihatkan pada Gambar 6.

A. IMPLEMENTASI RASPBERRY PI

Pada tahap implementasi, Raspberry Pi dapat membaca gambar kode QR yang aman yang dihasilkan oleh aplikasi siswa. Raspberry Pi akan melakukan hal-hal sebagai berikut.

1. Raspberry Pi mendeteksi kehadiran kode QR dari modul kamera dengan menggunakan *library* pyzbar.
2. *Library* pyzbar mendekode gambar kode QR menjadi format *string* yang telah dienkripsi menggunakan metode enkripsi RSA. Hasil pemindaian kode QR disediakan dalam format CSV untuk tujuan *debugging*.
3. Kode QR yang didekode berbentuk *ciphertext* dengan format “nama_pengguna#Nama_Siswa#hash_sandi#”.
4. *Password* tersebut akan didekripsi menggunakan kunci privat RSA dan Raspberry Pi akan meminta data nama pengguna dan sandi pada Firebase kemudian memverifikasinya. Jika verifikasi berhasil, *buzzer* akan memberi sinyal bahwa data telah dikonfirmasi.
5. Terakhir, Raspberry Pi mengirimkan penanda waktu ke Firebase untuk mencatat kehadiran siswa.

B. HASIL PENGUJIAN APLIKASI PADA ANDROID

Setelah implementasi, dilakukan pengujian kinerja dari program Android untuk mengukur kinerja sistem. Pengujian ini bertujuan untuk menentukan kecepatan proses autentikasi yang telah diterapkan. Dalam pengujian ini, kecepatan yang diukur merupakan kecepatan aplikasi Android dalam melakukan proses autentikasi serta jumlah energi yang diperlukan untuk melakukan proses autentikasi kehadiran siswa. Pengukuran ini dilakukan dengan bantuan perangkat lunak Android Studio

TABEL I
HASIL UJI WAKTU EKSEKUSI PROGRAM

Pengujian	Hasil		
	Tanpa Enkripsi	*RSA 1.024 bit*	*RSA 2.048 bit*
Waktu menampilkan kehadiran siswa oleh guru	1,66 s	1,66 s	1,66 s
Waktu menampilkan kode QR oleh siswa	0,97 s	1,33 s	1,76 s
Konsumsi energi untuk mengenkripsi data siswa	0,10 J	0,14 J	0,19 J

TABEL II
HASIL PEMBACAAN KODE QR

No	Pengujian	Hasil					
		5 cm	*6 cm*	*8 cm*	*11 cm*	*13 cm*	*14 cm*
1	Tanpa enkripsi	Tidak dapat dibaca	Dapat dibaca	Dapat dibaca	Dapat dibaca	Dapat dibaca	Dapat Dibaca
2	RSA 1.024 bit	Tidak dapat dibaca	Dapat dibaca	Dapat dibaca	Dapat dibaca	Dapat dibaca	Tidak dapat dibaca
3	RSA 2.048 bit	Tidak dapat dibaca	Dapat dibaca	Dapat dibaca	Dapat dibaca	Tidak dapat dibaca	Tidak dapat dibaca

untuk mengukur setiap proses presensi. Pengujian dilakukan sebanyak sepuluh kali, kemudian dihitung rata-rata dari proses pengukuran untuk mewakili data tersebut. Hasil pengujian ditampilkan dalam Tabel I. Hasil ini menentukan durasi waktu eksekusi program.

Tampak pada Tabel I bahwa tidak ada perbedaan waktu dalam menampilkan data kehadiran siswa pada aplikasi Android. Ketika program Android membuat kode QR terenkripsi, kode QR dengan enkripsi RSA 1.024 bit lebih lambat 0,36 s daripada tanpa enkripsi dan lebih cepat 0,42 s daripada enkripsi RSA 2.048 bit. Jumlah energi yang diperlukan untuk mengenkripsi kode QR menggunakan RSA 2.048 bit adalah 57,4 mJ lebih banyak dibandingkan dengan proses enkripsi menggunakan RSA 1.024 bit, dan membutuhkan energi 88,5 mJ lebih banyak dibandingkan dengan proses enkripsi tanpa menggunakan kode QR. Tabel I menunjukkan bahwa enkripsi tidak dilakukan pada panjang kritis lebih dari 2.048 karena Raspberry Pi tidak dapat membacanya.

C. PENGUJIAN PERANGKAT RASBERRY PI

Setelah tahap implementasi, dilakukan pengujian terhadap sistem yang telah dibuat dengan hasil pengujian kinerja program yang ditanamkan pada Raspberry Pi. Pengujian kinerja program dilakukan untuk mengetahui eksekusi program yang dilakukan dan menentukan enkripsi yang sesuai untuk mengamankan sistem presensi berbasis kode QR ini.

1) PENGUJIAN PEMBACAAN KODE QR OLEH RASBERRY PI

Pengujian pembacaan kode QR ini mengukur kemampuan Raspberry Pi untuk membaca kode QR berdasarkan panjang kunci enkripsi. Hasil pengujian pembacaan kode QR disajikan pada Tabel II. Tampak pada Tabel II bahwa jarak minimum untuk membaca kode QR adalah 6 cm untuk semua metode yang digunakan, sedangkan rentang jarak baca untuk setiap metode memiliki perbedaan 1 cm. RSA 2.048 bit memiliki rentang jarak baca terendah, yaitu 5 cm, pada jarak 6 cm sampai 11 cm.

TABEL III
PENGUJIAN KINERJA RASPBERRY PI DALAM MEMBACA KODE QR

No	Pengujian	Hasil		
		Penggunaan CPU (%)	*Memori (MB)*	*Waktu Eksekusi (s)*
1	Tanpa enkripsi	6	130,9	4,261
2	RSA 1.024 bit	28	132,9	6,285
3	RSA 2.048 bit	35	134,1	8,452

2) PENGUJIAN KINERJA RASBERRY PI DALAM MEMBACA KODE QR

Pengujian kinerja kode QR ini membandingkan kinerja program berdasarkan panjang kunci enkripsi yang diterapkan pada kode QR. Tabel III menunjukkan hasil pengukuran kinerja Raspberry Pi menggunakan tiga parameter, yaitu penggunaan CPU, penggunaan memori, dan waktu eksekusi. Proses enkripsi membuat CPU bekerja lebih keras dan lebih lama daripada ketika menggunakan RSA. Hal ini ditunjukkan oleh peningkatan penggunaan CPU, yaitu lebih dari 20% dibandingkan tanpa enkripsi, dan bertambah lamanya proses enkripsi 2 hingga 4 s saat menggunakan enkripsi RSA. Namun, selisih penggunaan memori antara eksekusi program dengan enkripsi dan tanpa enkripsi hanya berkisar 2 sampai 4 MB.

Pada penelitian sebelumnya [22], algoritma SM3+SM2 membutuhkan waktu lebih cepat dalam melakukan proses autentikasi, yaitu 2,841 kali/s, hampir tiga kali lebih cepat daripada sistem yang diusulkan. Hal ini disebabkan oleh perangkat keras yang digunakan pada penelitian sebelumnya lebih unggul daripada perangkat keras yang diusulkan dalam sistem ini [22]. Selain itu, algoritma SM3+SM2 juga lebih cepat daripada algoritma RSA dan SHA [22]. Namun, melalui pemeriksaan lebih lanjut, diketahui bahwa penggunaan perangkat keras tersebut tidak diperlukan karena biayanya yang berlebihan dan adanya banyak antarmuka yang tidak diperlukan [22].

IV. KESIMPULAN

Sistem autentikasi untuk sistem presensi berbasis kode QR menggunakan algoritma RSA dan *hash* telah dikembangkan untuk mengatasi kerentanan dalam aplikasi presensi siswa, yaitu siswa dapat memalsukan kode QR. Kode QR palsu ini dapat membuat siswa tercatat hadir tanpa datang ke kelas. Beberapa contoh standar keamanan untuk mengatasi masalah ini adalah penggunaan kriptografi, steganografi, dan penambahan lapisan keamanan pada pesan kode QR. Dalam makalah ini, diusulkan kode QR terenkripsi kunci publik menggunakan algoritma RSA dan ditambahkan fungsi SHA-1 untuk menjamin integritas pesan.

Beberapa parameter kinerja, termasuk waktu eksekusi, jarak pembacaan kode QR, dan kinerja program, diambil untuk mengukur kinerja sistem keamanan ini. Berdasarkan penelitian yang sudah dilakukan, implementasi sistem keamanan berbasis algoritma enkripsi kode QR RSA untuk sistem presensi dapat berfungsi dengan baik dan sesuai dengan desain yang telah direalisasikan. Implementasi sistem keamanan pada aplikasi kode QR untuk sistem presensi siswa dan algoritma RSA pada aplikasi berbasis kode QR diterapkan pada tiga perangkat, yaitu pada perangkat Android siswa, perangkat Android guru dan perangkat Raspberry Pi. Perangkat Android guru digunakan untuk menampilkan data kehadiran siswa yang tervalidasi, perangkat Android siswa digunakan untuk menghasilkan kode QR tersandikan, dan perangkat Raspberry Pi digunakan untuk melakukan pemindaian kode QR siswa.

Perangkat Raspberry Pi dan aplikasi presensi dapat melakukan proses autentikasi dengan baik. Perangkat Raspberry Pi dan aplikasi berbasis kode QR juga dapat berkomunikasi dengan server *cloud* Firebase dengan baik. Jarak pembacaan maksimum kode QR adalah 13 cm, sedangkan jarak pembacaan minimum adalah 6 cm. Jarak pembacaan optimal adalah antara 8-11 cm.

Penggunaan metode enkripsi RSA dan algoritma *hash* dapat mencegah siswa melakukan tindakan presensi ilegal, yang dapat dilakukan dengan menggunakan serangan pembangkitan kode QR palsu. Penambahan protokol keamanan pada sistem presensi dengan kode QR dapat membuat sistem lebih lambat. Pada setiap bagian, proses pelambatan tidak lebih dari 1 s dan secara keseluruhan, *delay* yang dihasilkan dalam proses masih kurang dari 12 s. Berdasarkan ITU-T G.1010, keterlambatan target untuk transfer data masif adalah 15 s, sehingga dapat dikatakan bahwa metode yang digunakan masih sesuai dengan target transfer data ITU-T.

Selain itu, penambahan protokol keamanan pada sistem presensi berbasis kode QR meningkatkan konsumsi penggunaan energi pada perangkat Android sebesar 0,09 J. Jika dikonversi ke mAh pada tengangan 5 V, nilainya adalah 0,0022 mAH, yang berarti hanya menggunakan tidak lebih dari 1,76% dari baterai 4.500 mAh setiap jamnya.

Kinerja sistem keamanan pada sistem presensi dengan kode QR dan algoritma RSA serta *hash* akan lebih presisi dan lebih cepat jika digunakan algoritma enkripsi simeteris seperti AES pada proses authentikasinya. Akan tetapi, penggunaan algoritma enkripsi simetris memerlukan modifikasi pada proses pendistribusian kunci menjadi lebih kompleks untuk setiap ruang kelas dan mata pelajaran guna meningkatkan sistem keamanannya.

KONFLIK KEPENTINGAN

Para penulis menyatakan bahwa artikel yang berjudul "Implementasi Sistem Keamanan Presensi Berbasis Kode QR Menggunakan Algoritma RSA dan *Hash*" ini bebas dari konflik kepentingan.

KONTRIBUSI PENULIS

Metodologi dan implementasi, Arif Indra Irawan; penulisan—persiapan awal, Maya Rahayu; penulisan—peninjauan dan penyuntingan, Istikmal; manajemen proyek dan pendanaan, Iman Hedi Santoso.

UCAPAN TERIMA KASIH

Karya ini hanya mungkin terwujud berkat dukungan keuangan dari Direktorat Penelitian dan Pengabdian Masyarakat Telkom University. Terima kasih juga diucapkan kepada semua individu, lembaga, dan perusahaan yang tidak dapat disebutkan satu per satu. Terima kasih atas bantuan Anda dalam menyelesaikan makalah ini dengan sukses.

REFERENSI

[1] V. Uzun, "QR-code based hospital systems for healthcare in Turkey," *2016 IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, 2016, hal. 71–76, doi: 10.1109/COMPSAC.2016.173.

[2] M.E. Çoban, B. Çubukçu, R. Yayla, dan U. Yüzgeç, "Raspberry Pi based robot application using QR code: QR-Robot," *2019 4th Int. Conf. Comput. Sci. Eng. (UBMK)*, 2019, hal. 119–123, doi: 10.1109/UBMK.2019.8907129.

[3] A.D.B. Sadewo, E.R. Widasari, dan A. Muttaqin, "Perancangan pengendali rumah menggunakan smartphone Android dengan konektivitas Bluetooth," *J. Pengemb. Teknol. Inf. Ilmu Komput.*, vol. 1, no. 5, hal. 415-425, Mei 2017.

[4] P. Tilala, A.K. Roy, dan M.L. Das, "Home access control through a smart digital locking-unlocking system," *TENCON 2017-2017 IEEE Region 10 Conf.*, 2017, hal. 1409–1414, doi: 10.1109/TENCON.2017.8228079.

[5] S. Tiwari, "An introduction to QR code technology," *2016 Int. Conf. Inf. Technol. (ICIT)*, 2016, hal. 39–44, doi: 10.1109/ICIT.2016.38.

[6] M.S. Akbar dkk., "Face recognition and RFID verified attendance system," *2018 Int. Conf. Comput. Electron. Commun. Eng. (iCCECE)*, 2018, hal. 168–172, doi: 10.1109/iCCECOME.2018.8658705.

[7] E. Susanto, D. Perdana, A.I. Irawan, dan R. Yasirandi, "Pengembangan sistem presensi menggunakan quick response code dinamis untuk Madrasah Aliyah Al Mukhlisin Bandung," *J. Rekayasa Elekt.*, vol. 15, no. 2, hal. 139-144, Agu. 2019, doi: 10.17529/jre.v15i2.13769.

[8] K.S.C. Yong, K.L. Chiew, dan C.L. Tan, "A survey of the QR code phishing: The current attacks and countermeasures," *2019 7th Int. Conf. Smart Comput. Commun. (ICSCC)*, 2019, hal. 1–5, doi: 10.1109/ICSCC.2019.8843688.

[9] A. Averin dan N. Zyulyarkina, "Malicious QR-code threats and vulnerability of blockchain," *2020 Glob. Smart Ind. Conf. (GloSIC)*, 2020, hal. 82–86, doi: 10.1109/GloSIC50886.2020.9267840.

[10] "Hubungan antara QR code dan dunia industri dan perdagangan," Pusdiklat Industri, 2020.

[11] T.M. Fernandez-Carames dan P. Fraga-Lamas, "A review on human-centered IoT-connected smart labels for the Industry 4.0," *IEEE Access*, vol. 6, hal. 25939–25957, 2018, doi: 10.1109/ACCESS.2018.2833501.

[12] L. Tan dkk., "Visual secret sharing scheme for color QR code," *2018 IEEE 3rd Int. Conf. Image Vis. Comput. (ICIVC)*, 2018, hal. 961–965, doi: 10.1109/ICIVC.2018.8492724.

[13] S. Liu, Z. Fu, dan B. Yu, "Rich QR codes with three-layer information using Hamming code," *IEEE Access*, vol. 7, hal. 78640-78651, Jun. 2019, doi: 10.1109/ACCESS.2019.2922259.

[14] N.V. Akhil, A. Vijay, dan D.S. Kumar, "QR code security using proxy re-encryption," *2016 Int. Conf. Circuit Power Comput. Technol. (ICCPCT)*, 2016, hal. 1-5, doi: 10.1109/ICCPCT.2016.7530286.

[15] A. Mendhe, D.K. Gupta, dan K.P. Sharma, "Secure QR-code based message sharing system using cryptography and steganography," *2018 1st Int. Conf. Secure Cyber Comput. Commun. (ICSCCC)*, 2018, hal. 188–191, doi: 10.1109/ICSCCC.2018.8703311.

[16] V. Malathi, B. Balamurugan, dan S. Eshwar, "Achieving privacy and security using QR code by means of encryption technique in ATM," *2017 2nd Int. Conf. Recent Trends Chall. Comput. Models (ICRTCCM)*, 2017, hal. 281–285, doi: 10.1109/ICRTCCM.2017.36.

[17] P.-Y. Lin dan Y.-H. Chen, "QR code steganography with secret payload enhancement," *2016 IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2016, hal. 1-5, doi: 10.1109/ICMEW.2016.7574744.

[18] Y.-M. Wang dkk., "Secured graphic QR code with infrared watermark," *2018 IEEE Int. Conf. Appl. Syst. Invent. (ICASI)*, 2018, hal. 690–693, doi: 10.1109/ICASI.2018.8394351.

[19] L.F. Freitas, A.R. Nogueira, dan M.E.V. Melgar, "Visual authentication scheme based on reversible degradation and QR code," *2020 4th World Conf. Smart Trends Syst. Secur. Sustain. (WorldS4)*, 2020, hal. 58-63, doi: 10.1109/WorldS450073.2020.9210412.

[20] M. Alajmi, I. Elashry, H.S. El-Sayed, dan O.S.F. Allah, "Steganography of encrypted messages inside valid QR codes," *IEEE Access*, vol. 8, hal. 27861–27873, Feb. 2020, doi: 10.1109/ACCESS.2020.2971984.

[21] I. Tkachenko dkk., "Two-level QR code for private message sharing and document authentication," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 3, hal. 571–583, Mar. 2016, doi: 10.1109/TIFS.2015.2506546.

[22] Y. Zhou, B. Hu, Y. Zhang, dan W. Cai, "Implementation of cryptographic algorithm in dynamic QR code payment system and its performance," *IEEE Access*, vol. 9, hal. 122362–122372, Agu. 2021, doi: 10.1109/ACCESS.2021.3108189.

[23] A.G. Konheim, *Computer Security and Cryptography*. Hoboken, AS: John Wiley & Sons, 2007.

[24] Y. Zhao, Y. Li, dan S. Wang, "Asymmetric deep hashing for person re-identifications," *Tsinghua Sci. Technol.*, vol. 27, no. 2, hal. 396–411, Apr. 2022, doi: 10.26599/TST.2021.9010014.

[25] T. Kleinjung dkk., "Factorization of a 768-Bit RSA Modulus," dalam *Advances in Cryptology – CRYPTO 2010*, T. Rabin, Ed., Heidelberg, Jerman: Springer Berlin, 2010, hal. 333–350, doi: 10.1007/978-3-642-14623-7_18.