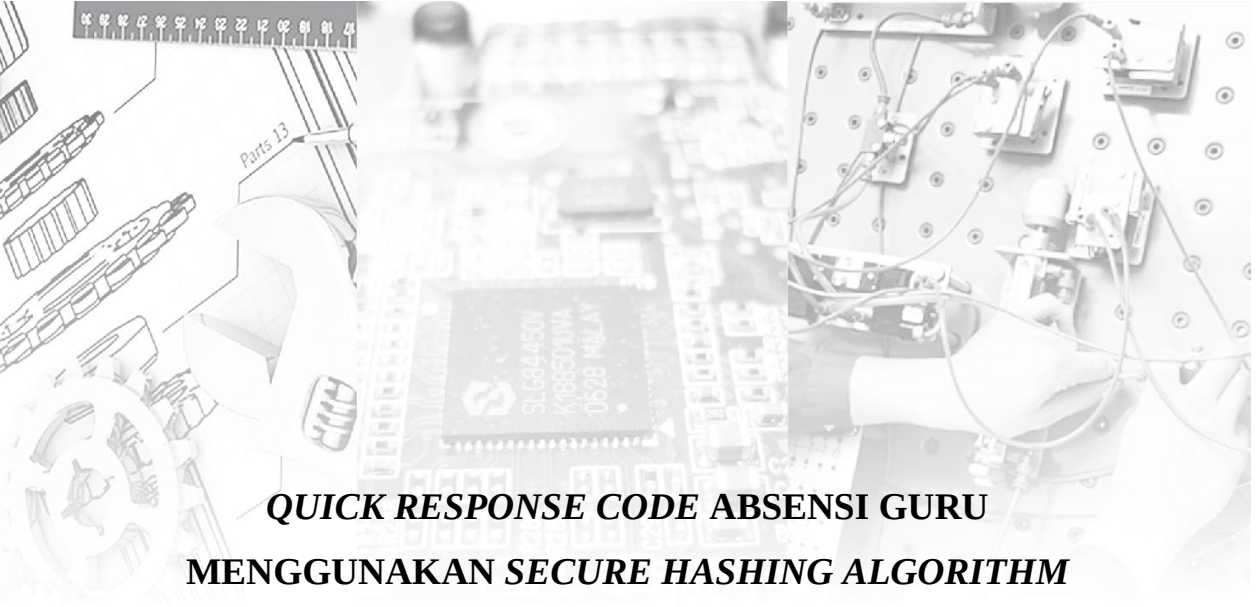


QUICK RESPONSE CODE ABSENSI GURU MENGGUNAKAN *SECURE HASHING ALGORITHM* (SHA)

Agriyanto Asiking [1], Asmaul Husnah N[2], Irma Surya Kumala Idris [3]
Teknik Informatika, Universitas Ichsan Gorontalo
faizalluly@gmail.com [1], asmaulhusnel@gmail.com[2] , mhaladp@gmail.com [3]

Abstrak

Sistem Absensi guru yang diterapkan di sekolah masih dilakukan secara manual, yaitu guru menandatangani buku absen yang telah disediakan. Hal ini dikhawatirkan dapat meningkatkan potensi penyebaran COVID 19 dikarenakan menggunakan peralatan absensi yang sama. Berdasarkan permasalahan tersebut penelitian absensi akan dibuat menggunakan teknologi antara *Quick Response Code* yang menggunakan *Secure Hash Algorithm* (SHA) dan Smartphone android sehingga mengurangi kontak fisik atau penggunaan benda yang disentuh oleh banyak orang secara bergantian. Penelitian ini mengimplementasikan algoritma kriptografi SHA-256 untuk pembuatan *Quick Response Code* absensi. Hasil enkripsi dari SHA-256 akan dikombinasi dengan algoritma BCRYPT untuk menghindari serangan decode hash SHA-256. Pengamanan *Quick Response Code* dengan menggunakan enkripsi SHA-256 lebih optimal dengan mengkombinasikan fungsi BCRYPT pada Message yang telah dienkripsi SHA-256, sehingga menghindari serangan decode hash SHA-256

Kata Kunci: *Quick Response Code*; Absensi; *Secure Hash Algorithm*

ABSTRACT

The teacher attendance system applied in schools is still done manually, namely the teacher signs the attendance book that has been provided. This is feared to increase the

potential for the spread of COVID 19 due to using the same attendance equipment. Based on these problems, attendance research will be made using technology between the Quick Response Code that uses the Secure Hash Algorithm (SHA) and the Android Smartphone so as to reduce physical contact or the use of objects that are touched by many people in turn. This research implements the SHA-256 cryptographic algorithm for making Quick Response Code attendance. The encryption results from SHA-256 will be combined with the BCRYPT algorithm to avoid SHA-256 hash decoding attacks. Quick Response Code security using SHA-256 encryption is more optimal by combining the BCRYPT function on messages that have been encrypted SHA-256, thus avoiding SHA-256 hash decode attacks.

Keywords*: Quick Response Code, attendance, Secure Hash Algorithm*

A. PENDAHULUAN

Absensi guru merupakan evaluasi awal untuk mengetahui hadir atau tidaknya seorang guru di Sekolah. Hal ini dilakukan agar dapat menjaga kedisiplinan guru dalam mengemban tanggung jawab sebagai pengajar. Kedisiplinan guru menjadi sangat berarti bagi keberhasilan seorang guru dalam mengajar dan kemajuan sekolah serta meningkatkan prestasi belajar siswa (Nashir, 2016). Sehingga dapat disimpulkan bahwa absensi guru dapat dijadikan sebagai tolak ukur awal dalam menilai bagaimana seorang guru disiplin terhadap waktu dan tanggung jawabnya untuk menghadiri jam kerja yang telah ditentukan oleh pihak Sekolah.

Sistem absensi guru yang diterapkan di Sekolah masih bersifat manual, yaitu guru datang mengisi buku absensi dengan menandatangani kolom absensi yang tersedia pada buku absen. Adapun buku absen dan alat tulis yang digunakan sudah disiapkan lebih dulu, sehingga semua guru menggunakan alat tersebut secara bergantian. Hal ini dikhawatirkan dapat meningkatkan potensi penyebaran virus COVID-19 yang masih menjadi pandemi hingga saat ini. Perlu diketahui bahwa potensi penyebaran virus COVID-19 dapat dicegah dengan menghindari kontak

fisik atau penggunaan suatu benda umum yang disentuh oleh orang banyak.

Berdasarkan fenomena yang telah dijelaskan diatas maka penulis akan membuat aplikasi absensi dengan memanfaatkan teknologi antara *Quick Response Code* yang menggunakan *Secure Hash Algorithm* (SHA) dan Smartphone Android sehingga mengurangi kontak fisik atau penggunakan benda yang disentuh oleh orang banyak secara bergantian.

Quick Response Code atau QR Code merupakan hasil perkembangan teknologi pada smartphone yang berbentuk matrik 2 dimensi dengan pembacaan yang cepat dan kapasitas penyimpanan karakter yang lebih besar. *Quick Response Code* saat ini telah banyak digunakan di dunia industri, perdagangan, dan dunia Pendidikan (Supendi, et al., 2019). *Quick Response Code* merupakan pengembangan dari barcode satu dimensi dan merupakan salah satu tipe dari barcode yang dapat dibaca dengan menggunakan kamera handphone (Sujarwo, et al., 2020). *Quick Response Code* dapat diterapkan pada absensi guru karena mudah digunakan pada smartphone android namun memiliki keamanan yang tinggi karena dikombinasikan dengan Secure Hash *Algorithm* (SHA).

Secure Hash Algorithm (SHA) merupakan salah satu algoritma kryptografi yang digunakan untuk melakukan *encrypt* pesan atau karakter. Fungsi *hash* pada *Secure Hash Algorithm* (SHA) hanya bekerja satu arah, sehingga pesan atau karakter yang telah diubah menjadi pesan *digest* tidak dapat dikembalikan menjadi pesan atau karakter semula. Dua pesan atau karakter yang berbeda akan menghasilkan nilai hash yang berbeda pula (Damanik, 2017). Maka untuk menghindari pemalsuan *Quick Response Code* yang saat ini mudah dilakukan karena aplikasi *generating*

Quick Response Code sudah tersedia pada smartphone, penulis tertarik untuk menggunakan *Secure Hash Algorithm* (SHA) pada *Quick Response Code* absensi guru agar *QR Code* absensi guru tidak dapat dipalsukan.

Berdasarkan Latar belakang yang ada, maka penulis membuat penelitian dengan judul **"*Quick Response Code Absensi Guru Menggunakan Secure Hashing Algorithm (SHA)*".** Dengan tujuan agar absensi guru mejadi lebih valid karena penggunaan *Quick Response Code* dan *Secure Hash Algorithm* (SHA) yang memiliki tingkat keamanan yang tinggi, serta meminimalisir kontak fisik atau penggunaan benda yang disentuh oleh orang banyak sehingga mengurangi potensi penyebaran virus COVID-19.

B. METODE

Secure Hash Algorithm 256 (SHA-256) adalah sebuah kriptografi fungsi hash yang dirancang oleh National Security Agency (NSA) dan dipublikasikan oleh National Institute of Standart and Technology (NIST) sebagai sebuah Federal Information Processing Standart (FIPS) oleh U.S. Ada empat algoritma untuk keamanan fungsi hash yaitu SHA-0, SHA-1, SHA-2, dan SHA-3. NIST memperbaharui SHA-2, dengan panjang output (256 atau 512- bit di atas 160-bit pada SHA-1) dan perbedaan-perbedaan pada SHA ini merupakan besar pesan yang ada pada proses komputasi (Ibrahim, 2015).

SHA (Algoritma keamanan fungsi hash) merupakan algoritma enkripsi fungsi hash yang dapat digunakan untuk menghasilkan penggambaran konsolidasi dari sebuah data teks yang disebut sebuah proses pesan. SHA-256 dan SHA-512 adalah fungsi hash dengan kapasitas terbaru dengan panjang 32-bit dan 64-bit kata secara terpisah.

Kedua fungsi hash ini dalam proses matematisnya menggunakan penjumlahan karakter yang berbeda dan ditambah dengan konstanta substansi. Meski demikian, struktur keduanya pada dasarnya tidak jauh berbeda, perbedaannya hanya terletak pada jumlah putaran saja (Sukhbir, 2017).

Adapun proses atau tahapan pada algoritma SHA-256 adalah sebagai berikut (Panjaiatan, et al., 2020):

1. Message Padding

   Pada tahap pertama ini, pesan berupa binary disisipkan dengan angka 1 dan ditambahkan bit-bit pengganjal, yakni angka 0 hingga panjang pesan kongruen dengan 448 modulo 512. Panjang pesan asli ditambah sebagai angka biner 64 bit. Maka panjang pesan sekarang menjadi kelipatan 512 bit.

2. Parsing

   Pada proses ini, pesan yang telah dipadding kemudian dibagi menjadi N buah blok 512 bit : $M(1)$, $M(2)$, … $M(n)$

3. Message Expansion

   Masing-masing blok 512 bit tadi dipecah menjadi 16 word 32 bit : $M0$ (i) , $M1$ (i) , … $M15$ (i) Kemudian akan diperluas menjadi 64 word yang diberli label W0, W1,...W63.

4. Message Compression

   Masing-masing dari 64 word yang diberi label W0, W1,...W64 tadi diproses denga algoritma fungsi hash SHA-256. Dalam proses tersebut, inti utama dari algoritma SHA-256 adalah membuat 8 variabel yang diberikan nilai awal L0-L7. Nilai awal tersebut adalah sebagai berikut:

Tabel 1. Nilai Awal Variabel SHA-256

L0	a	6A09E667	L4	E	510E527F
L1	b	BB67AE85	L5	F	9B05688C
L2	c	3C6EF372	L6	G	IF83D9AB
L3	d	A54FF53A	L7	H	5BE0CD19

5. Kemudian dilakukan perhitungan sebanyak 64 kali pitaran untuk setiap blok. Delapan variabel yang diberikan pada nilai awal berupa L0 sampai dengan L7 asumsikan menjadi nilai A,B,C,D,E,F,G, dan H nilainya terus berganti selama perputaran dengan rumus sebagai berikut:

   $T1 = \hbar + s1 + CH + K[t] + W[t]$ $\quad T2 = s0 + MAJ$

   $\hbar = g,\ g = f,\ f = e,\ e = d + T1,\ d = c,\ c = b,\ b = a,\ a = T1 + T2$

   Keterangan :

   $s0 = (a \ggg 2) \oplus (a \ggg 13) \oplus (a \ggg 22)$, $s1 = (e \ggg 6) \oplus (e \ggg 11) \oplus (e \ggg 25)$, $CH = (e \& f) \oplus ((\neg e) \& g))$

   $MAJ = (a \& b) \oplus (a \& c) \oplus (b \& c)$

   Nilai akhir hash adalah sebagai berikut:

   $L0 = L0 + a,\ L1 = L1 + b$ $\quad L2 = L2 + c,\ L3 = L3 + d$

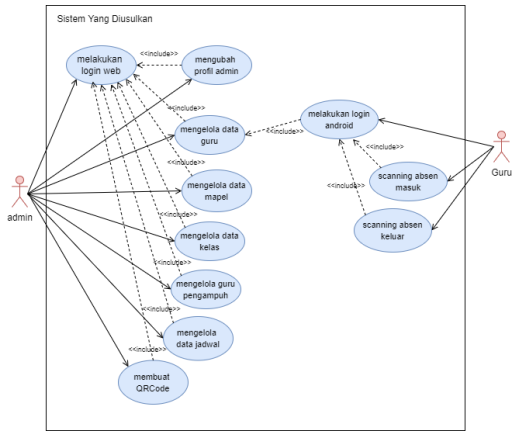
   $L4 = L4 + e,\ L5 = L5 + f$ $\quad L6 = L6 + g,\ L7 = L7 + \hbar$

Maka, MD yang didapatkan adalah hasil akhir penjumlahan yang disusun secara memanjang (Panjaiatan, et al., 2020).

C. HASIL DAN PEMBAHASAN

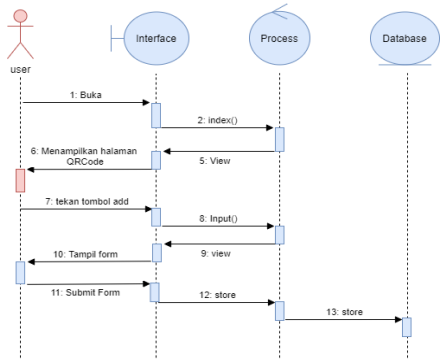
a. Desain Sistem

Aplikasi yang akan dibuat didesain menggunakan bahasa pemodelan visual dengan metode UML (Unified Modelling Language) sehingga dapat dijabarkan dalam bentuk yang baku dan mudah dimengerti:

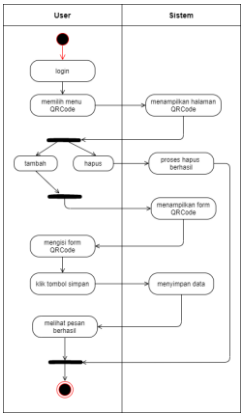


Gambar 1. Sistem yang diusulkan

Berdasarkan gambar 1 dapat dideskripsikan bahwa sistem yang di usulkan terdapat 2 Aktor, yaitu admin, guru. *Behavior* admin pada *use case* yaitu melakukan login web yang selanjutnya setelah login dapat mengubah profil admin, mengelola data guru, mengelola data mapel, mengelola data kelas, mengelola data guru pengampuh, mengelola data jadwal dan membuat *Quick Response Code*. Kemudian aktor Guru dapat melakukan scanning *Quick Response Code*. Aktor guru harus melakukan login android terlebih dahulu agar dapat melakukan scanning absen masuk dan absen keluar.

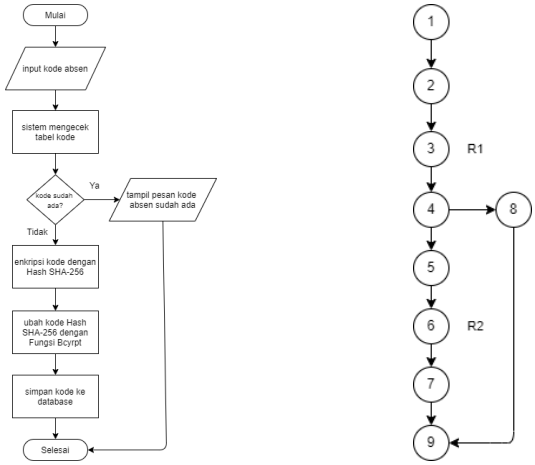


Gambar 2. Activity Diagram Buat *Quick Response Code*



Gambar 3. Sequence Diagram Buat *Quick Response Code*

b. Pengujian Sistem



Gambar 4. *Flowchart* dan *Flowgraph* Proses Pembuatan QR Code

Dari Flowgraph tersebut didapatkan:

Diketahui: Region (R) = 2 Node (N) = 9

Edge (E) = 9 Predikat Node (P) = 1

Rumus: V(G) = E - N + 2 dan V(G) = P + 1

V(G) = 9 - 9 + 2 = 2

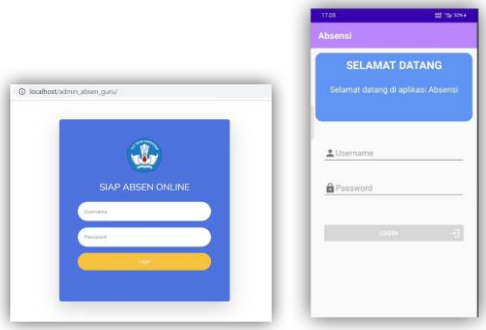
V(G) = 1 + 1 = 2

Basis Path :

Path 1 : 1-2-3-4-5-6-7-9

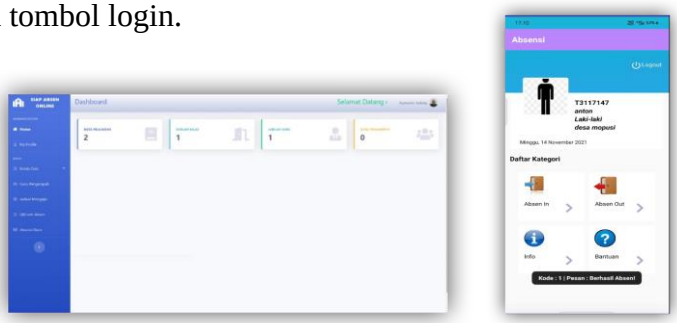
Path 2 : 1-2-3-4-8-9

c. Pengujian Sistem



Gambar 5. Tampilan Login Admin dan Guru

Halaman pada gambar 5 bagian pertama digunakan untuk login admin dengan memasukkan username dan password kemudian menekan tombol login, sedangkan gambar 5 bagian kedua digunakan oleh guru untuk login guru dengan memasukkan username dan password dan menekan tombol login.



Gambar 6. Tampilan Halaman Utama Admin dan Guru

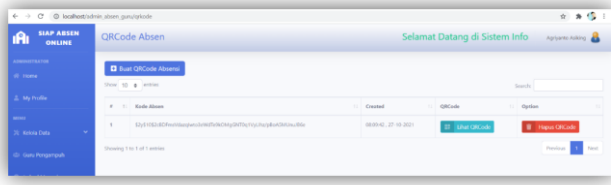
Gambar 6 adalah halaman utama saat admin berhasil login, dan halaman utama saat guru berhasil melakukan login.

Gambar 7. Tampilan Buat *Quick Response Code*

Halaman diatas digunakan untuk membuat *Quick Response Code* absensi oleh admin.

Gambar 8. Tampilan Daftar Guru

Halaman ini menampilkan data guru yang telah diinput oleh admin dan dapat diedit atau dihapus.



Gambar 9. Tampilan Data *Quick Response Code*

Halaman ini menampilkan Quick Response Code yang sudah dibuat oleh admin. Quick Response Code dapat dilihat dengan menekan tombol lihat qrcode pada kolom. QRCode juga bisa dihapus.

d. Proses Enkrip SHA-256

Pada proses ini penulis akan melakukan pembahasan pada proses Enkrip SHA-256 kemudian di amankan lagi dengan Bckrypt.

1) Langkah pertama yaitu input plain teks

   Plain teks adalah sebuah kata atau karakter yang akan kita enkrip dengan SHA256 dan Bcrypt, Plain teks yang akan di buat oleh penulis yaitu : tes

2) Pada tahap ini pesan yang dimasukan oleh penulis tadi akan di ubah ke dalam bentuk angka biner agar bisa ke tahap selanjutnya.

   t = 01110100 e = 01100101 s = 01110011

   Setelah di ubah ke angka biner maka pesan “tes” tadi menjadi

   M=011101000110010101110011

   Diketahui panjang pesan = 24 bit

3) Message Padding

   Padding pesan yang telah di ubah ke angka biner tadi dengan cara menambahkan bit 1 dan sisanya bit nol hingga pesan sepanjang 512 bit. Untuk menemukan berapa bit nol yang akan ditambahkan maka laukan dengan rumus :

   $I + 1 + k = 448 \ mod \ 512$ $k = 448\text{-}25 \ mod \ 512$

   $24+1+k = 448 \ mod \ 512$ $k = 423$

   Karena K = 423 maka banyaknya bit 0 yang akan ditambahkan adalah sebanyak 423 bit.

4) *Parsing*

   Pesan yang telah dipadding menghasilkan blok pesan 512 bit selanjutnya adalah melakukan pembagian setiap blok 512 bit menjadi 16 buah word 32 bit.

5) Message Schadule

Tahap ini adalah memperluas memperluas masing-masing 16 buah word yang telah diparsing menjadi 64 buah 32 bit word.

6) Inisialisasi Variabel dan Konstanta

Tahap kita hanya perlu menuliskan variabel awal pada fungsi SHA-256 kemudian kita inisialisasikan, yaitu sebagai berikut:

a = H0(0) = 6A09E667, b = H0(1) = BB67AE85

c = H0(2) = 3C6EF372, d = H0(3) = A54FF53A

e = H0(4) = 510E527F, f = H0(5) = 9B05688C

g = H0(6) = 1F83D9AB

7) Hash Computation

Dalam proses ini dilakukan perhitungan nilai a sampai h sebanyak 64 kali putaran.

8) *Compute intermediate Hash Value + Initial Hash Value*

Setelah didapat hasil ke 64 dalam proses *Hash Computation* kemudian dilakuan proses penjumlahan hasil ke-64 dengan nilai hash value.

9) Penggabungan H0 – H7

Tahap ini kita hanya perlu melakukan penggabungan hasil dari penjumlahan ke-64 dan hash value.

10) Nilai Hash

Dengan seluruh proses yang telah dilalui didapatlah nilai hash SHA-256 dari pesan tes:

ce0f6c28b5869ff166714da5fe08554c70c731a335ff9702e38b00f81ad348c6

d. Proses Bcrypt

Pada proses ini penilis akan melakukan pengaman hasil dari hash SHA-256 di atas degan BCRYPT untuk menghindari serangan decode SHA-256.

1) plain teks password

   Pada tahap ini penulis memasukan plain teks password yang telah di hash menggunakan SHA-256 yaitu :

   ce0f6c28b5869ff166714da5fe08554c70c731a335ff9702e38b00f81ad348c6

2) Cost

   Memasukan jumlah hashing acak yang dijalankan dengan jumlah minimal cost 10 samapai 30

3) Salt

   Menambahkan karakter acak yang akan di tambahkan di dalam proses enkripsi sebuah key. Salt berfungsi melapisi dan menjaga keamanan sebuah data yang telahdi enkripsi.

4) Proses *Hashing*

   melakukan proses plaintext yang telah di inputkan agar terbentuk ke dalam algoritma hash yaitu algoritma Bcrypt yang di gunakan oleh penulis dengan besaran 192 bit proses hash yang dihasilkan.

5) Hasil bcrypt, cost, salt dan hash

   H asil dari proses bcrypt, cost, salt dan hash yaitu:

   Bcrypt = $2y$ Cost =10$

   Salt =aSjYS2ng96ouyDPDEkA1R

   Hash =eQar2UoRWOlwJwVuc2.Fe7/pbcGbOFva

6) Hasil akhir

Hasil dari pesan hash SHA-256 yang di amankan dengan BCRYPT yaitu:

$2y$10$aSjYS2ng96ouyDPDEkA1ReQar2UoRWOlwJwVuc2.Fe7/pbcGbOFva

Gambar 10. Tampilan Hasil Proses SHA-256 dan BCRYPT

D. PENUTUP

Simpulan dan Saran

Berdasarkan penelitian yang dilakukan implementasi SHA-256 pada *Quick Response Code* absensi dan pembahasan yang sudah diuraikan sebelumnya, kesimpulan yang diperoleh yaitu Sistem yang dirancang mampu melakukan generate *Quick Response Code* dan Scan *Quick Response Code* untuk absensi melalui smartphone android sehingga meminimalisir kontak fisik atau penggunaan benda yang disentuh oleh orang banyak sehingga mengurangi potensi penyebaran virus COVID-19. Pengamanan *Quick Response Code* dengan menggunakan enkripsi SHA-256 lebih optimal dengan mengkombinasikan fungsi BCRYPT pada Message yang telah dienkripsi SHA-256, sehingga menghindari serangan decode hash SHA-256.

Setelah melakukan penelitian dan pembuatan suatu program untuk mengoptimalkan keamanan *Quick Response Code* absensi dengan menggunakan SHA-256 dan dikombinasi dengan Fungsi Hash BCRYPT, maka diharapkan pengembangan aplikasi untuk menambahkan fitur Geolocation untuk mengukur radius user agar absensi hanya dapat

dilakukan di wilayah sekolah. Penelitian kedepannya diharapkan menggunakan algoritma kryptografi yang berbeda dan kombinasi yang berbeda juga untuk menghasilkan penelitian yang lebih baik lagi.

DAFTAR PUSTAKA

Damanik, R. (2017). Pengkodean Pesan Teks Dengan Proses Penerapan Algoritma Kriptografi Secure Hash Algorithm (SHA). Jurnal Informatika Kaputama (JIK), 1(1), 48-57.

Ibrahim, R. K., Kadhim, R. A. J., & Alkhalid, A. S. H. (2015, September). Incorporating SHA-2 256 with OFB to realize a novel encryption method. In 2015 World Symposium on Computer Networks and Information Security (WSCNIS) (pp. 1-6). IEEE.

Nashir, A. (2016). Pengaruh kedisiplinan guru terhadap prestasi belajar. TARBAWI: Jurnal Pendidikan Agama Islam, 1(1), 21-28.

Panjaitan, Z., Ginting, E. F., & Yusnidah, Y. (2020). Modifikasi SHA-256 dengan Algoritma Hill Cipher untuk Pengamanan Fungsi Hash dari Upaya Decode Hash. Jurnal SAINTIKOM (Jurnal Sains Manajemen Informatika dan Komputer), 19(1), 53-61.

Sujarwo, Y. A., & Ratnasari, A. (2020). Aplikasi Reservasi Parkir Inap Menggunakan Metode Fishbone Diagram dan QR-Code. Jurnal Sisfokom (Sistem Informasi dan Komputer), 9(3), 302-309.

Sukhbir, C. & S-S, M. (2017). Crypto currencies for digital currency using cipher text and SHA256. Journal of Advanced Research in Dynamical and Control Systems, 2017:530–534.

Supendi, Y., Supriadi, I., & Isto, A. A. (2019, November). Pemanfaatan Teknologi QR-Code Pada Sistem Presensi Mahasiswa Berbasis Mobile. In SEMINAR NASIONAL APTIKOM (SEMNASTIK) 2019 (pp. 550-558).