SMART PRESENSI MENGGUNAKAN QR CODE DENGAN SECURE HASH ALGORITHM 2 (SHA-2)

Ivan Wijaya [1] Dyah Erny Herwindiati [2] Janson Hendryli [3]

[1] Teknik Informatika, FTI, Universitas Tarumanagara
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia
ivan.535170068@stu.untar.ac.id [1] dyahh@fti.untar.ac.id [2] jansonh@fti.untar.ac.id [3]

ABSTRACT

This research was motivated by problems during the process of attendance at the Mid-Semester Examination (UTS) and Final Semester Examination (UAS). The problem begins when the student attendance process is still manual and not online or connected to the database. This study aims to determine the online student attendance system so that exam supervisors no longer give signatures on student exam cards so that the exam runs well and efficiently. The data that will be used are student data taken when the administrator fills in the student detail data. The system design made is by using a QR Code Scanner and data encryption using the SHA 512 method. Using this application system includes the QR Code Scanning process, inserting a QR Code into the student exam card. Then the smart phone application is used as a QR Code scanner. The test used to give a test the feasibility of the system in this study is to use system error level testing, black box testing and the presence of QR Code testing in certain cases. In the testing that has been done, it can be concluded that the Smart Presensi application can make it easier to do attendance and student attendance data recapitulation because the system application is directly connected to the database.

Key words

Encryption, QR Code, Scanning, Secure Hash Algorithm 2 (SHA – 2), *Smart Presensi*

1. Pendahuluan

Pada perkembangan teknologi yang semakin maju dan pesat saat ini sangat berpengaruh pada kemudahan dalam aktivitas sehari-hari dalam berbagai bidang, tidak terkecuali dalam bidang pendidikan. Namun masih banyak sistem dalam dunia pendidikan Indonesia yang masih menggunakan cara tradisional dalam menjalankan salah satunya adalah sistem presensi. Presensi merupakan salah satu faktor penting dalam dunia pendidikan. Sistem presensi adalah sistem yang mencatat kehadiran dari setiap murid atau mahasiswa yang mengikuti kegiatan belajar. Walau dengan perkembangan teknologi yang semakin maju di dunia pendidikan, masih banyak juga sistem pendidikan yang masih tidak dapat memanfaatkan teknologi tersebut khususnya dalam sistem presensi.

Salah satu teknologi yang dapat digunakan dalam sistem presensi adalah *QR Code* (*Quick Response Code*). *QR Code* adalah suatu jenis kode matriks atau kode batang dua dimensi yang mempunyai berbagai kelebihan yaitu dari media penyimpanan yang besar untuk menampung konten di dalam *QR Code*, dari alphanumerik hingga huruf kanji. Selain dari penyimpanan yang luas, *QR Code* juga memiliki kelebihan lainnya yaitu mempunyai keamanan yang lebih baik karena hanya dapat dilihat dengan citra optik mesin dan tidak dapat dilihat langsung dengan mata telanjang.

Banyaknya teknologi yang telah beredar saat ini membuat keamanan yang dimiliki oleh *QR Code* semakin mudah dibaca dengan adanya aplikasi pada *smartphone* yang mudah untuk didapatkan. Untuk mengatasi pelemahan keamanan dari *QR Code* tersebut dapat diatasi dengan menggunakan fungsi hash. Salah satu fungsi hash yang terkenal adalah *Secure Hash Algorithm 2* (SHA – 2) dengan keluaran yang mempunyai panjang 512 bits atau biasa disebut dengan SHA – 512. Fungsi hash SHA – 512 merupakan fungsi hash yang memiliki arsitektur desain perhitungan yang berbeda dengan fungsi hash lainnya walau dengan bagian dari keluarga fungsi hash *Secure Hash Algorithm 2* (SHA – 2).

Beberapa hal inilah yang mendorong pemikiran mengenai membangun sistem yang dapat melakukan presensi mahasiswa secara efektif dan efisien. Dengan penggunaan *QR Code* dan SHA – 512 membuat data mahasiswa akan menjadi lebih aman karena data tidak akan dapat dibaca oleh mata dan bahkan oleh optik mesin.

2. Landasan Teori

2.1 Presensi

Presensi adalah dokumen yang mencatat daftar hadir setiap anggota suatu organisasi. Catatan daftar hadir presensi dapat berupa daftar hadir biasa yang terdapat tanda tangan setiap anggota atau dapat pula berbentuk yang lainnya seperti kartu dengan RFID bahkan dengan *biometric security* yang menggunakan *face recognition* dan *voice recognition*.

Secara umum jenis presensi dapat dikelompokkan menjadi dua jenis, yaitu :

1. Presensi manual
   Presensi manual adalah cara pencatatan kehadiran dengan cara menggunakan pena dan kertas atau biasa juga disebut menggunakan tanda tangan.
2. Presensi non manual
   Presensi non manual adalah suatu cata pencatatan kehadiran dengan menggunakan sistem terkomputerisasi, bisa dengan kartu dengan barcode, RFID, sidik jari ataupun dengan memasukkan nim dan sebagainya

2.2 *QR Code*

QR Code adalah suatu jenis kode matriks atau barcode dua dimensi yang berasal kata "*Quick Response Code*". *QR Code* dikembangkan oleh Denso Corporation pada tahun 1994. *QR Code* memiliki dua jenis, yaitu *Static QR Code* dan *Dynamic QR Code*. *Static QR Code* adalah jenis *QR Code* yang berisi tautan atau teks yang tetap dan menyebabkan konten dari *QR Code* tidak dapat diubah. Sedangkan, *Dynamic QR Code* adalah *QR Code* yang berisi tautan singkat yang kemudian akan dialihkan ke halaman web yang lain dan menyebabkan konten dari *QR Code* dapat diubah dan dapat digunakan kembali. Untuk contoh *QR Code* dapat dilihat pada gambar dibawah ini.

Gambar 1 Contoh *QR Code* "Ivan Wijaya, 535170068"

QR Code dapat menyimpan informasi secara *horizontal* dan *vertical*, oleh karena itu *QR Code* dapat menampung informasi atau konten yang lebih banyak daripada kode batang.

Tabel 1 Kapasitas *QR Code*

Tipe Data	Maximum Karakter
Numerik	7.089
Alphanumerik	4.296
Biner	2.953
Kanji	1.817

Selain dari kapasitasnya yang besar, *QR Code* juga mempunyai kelebihan lainnya, yaitu :

1. Kapasitas isi konten yang besar.
2. Ukuran yang relatif kecil.
3. Dapat menyimpan tipe data kanji dan kana
4. Ketahanan dari kotoran dan kerusakan hingga 30%
5. Dapat dibaca dari berbagai arah
6. Dapat menyatukan struktur dari beberapa *QR Code* menjadi 1 *QR Code*. [1]

2.3 *Secure Hash Algorithm – 512* (SHA – 512)

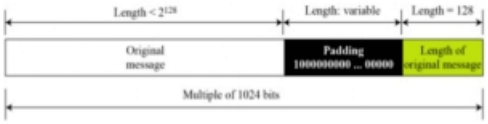
Secure Hash Algorithm – 512 merupakan bagian dari keluarga fungsi hash *Secure Hash Algorithm 2* (SHA-2) yang dirancang oleh Badan Keamanan Nasional Amerika Serikat (NSA) dan pertama kali dipublikasikan pada tahun 2001.

Tabel 2 Fungi Hash *Secure Hash Algorithm 2* (SHA – 2)

Algorithm	*Output size*	*Internal state size*	*Block Size*	*Operator*	*Security against collision attacks*
SHA-224	224	256 (8X64)	512	And, Xor, Rot, Add (mod 2^{32}), Or, Shr	112
SHA-256	256				128
SHA-384	384	512 (8X64)	1024	And, Xor, Rot, Add (mod 2^{64}), Or, Shr	192
SHA-512	512				256
SHA-512/224	224				112
SHA-512/256	256				128

Pada tabel diatas, fungsi hash SHA – 512 merupakan fungsi hash yang menghasilkan *message digest* dengan ukuran paling besar dengan 512 bit dan panjang blok 1024 bit. Hal ini menjadikan SHA – 512 lebih aman dibandingkan dengan keluarga fungsi hash SHA – 2. Dalam fungsi hash ini terdapat beberapa proses untuk mendapatkan hasil *hash code* dari fungsi SHA – 512

1. Mengubah teks input menjadi binary, lalu menambahkan padding dan length hingga menjadi format teks dengan ukuran kelipatan 1024 bit seperti pada gambar dibawah ini.



Gambar 2 *format input*

2. Mencari Qword yaitu dengan membagi format teks menjadi beberapa blok dengan ukuran 1024 bit, lalu satu blok dipecah kembali menjadi 16 Qword dengan ukuran 64 bit. Setelah mendapatkan 16 Qword maka

selanjutnya adalah mencari Qword ke-16 sampai 79 dengan menggunakan rumus pada persamaan 1. [2]

$$W_t = \sigma_1^{512}(W_{t-2}) + W_{t-7} + \sigma_0^{512}(W_{t-15}) + W_{t-16} \quad (1)$$

$$\sigma_0^{512}(x) = ROTR^1(x) \oplus ROTR^8(x) \oplus SHR^7(x)$$

$$\sigma_1^{512}(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x)$$

Keterangan:
$\oplus$ = XOR
$ROTR^n(x)$ = Circular Right Shift (Rotasi) dari argumen x dengan n bits
$SHR^n(x)$ = Left Shift (Rotasi) dari argumen x dengan n bits

3. Inisialisasi 80 konstan dan 8 buffers awal. Untuk inisialisasi konstan dapat dilihat pada gambar 3 dan inisialisasi buffers awal dapat dilihat pada gambar 4 [3]

```
428a2f98d728ae22 7137449123ef65cd b5c0fbcfec4d3b2f e9b5dba58189dbbc
3956c25bf348b538 59f111f1b605d019 923f82a4af194f9b ab1c5ed5da6d8118
d807aa98a3030242 12835b0145706fbe 243185be4ee4b28c 550c7dc3d5ffb4e2
72be5d74f27b896f 80deb1fe3b1696b1 9bdc06a725c71235 c19bf174cf692694
e49b69c19ef14ad2 efbe4786384f25e3 0fc19dc68b8cd5b5 240ca1cc77ac9c65
2de92c6f592b0275 4a7484aa6ea6e483 5cb0a9dcbd41fbd4 76f988da831153b5
983e5152ee66dfab a831c66d2db43210 b00327c898fb213f bf597fc7beef0ee4
c6e00bf33da88fc2 d5a79147930aa725 06ca6351e003826f 142929670a0e6e70
27b70a8546d22ffc 2e1b21385c26c926 4d2c6dfc5ac42aed 53380d139d95b3df
650a73548baf63de 766a0abb3c77b2a8 81c2c92e47edaee6 92722c851482353b
a2bfe8a14cf10364 a81a664bbc423001 c24b8b70d0f89791 c76c51a30654be30
d192e819d6ef5218 d69906245565a910 f40e35855771202a 106aa07032bbd1b8
19a4c116b8d2d0c8 1e376c085141ab53 2748774cdf8eeb99 34b0bcb5e19b48a8
391c0cb3c5c95a63 4ed8aa4ae3418acb 5b9cca4f7763e373 682e6ff3d6b2b8a3
748f82ee5defb2fc 78a5636f43172f60 84c87814a1f0ab72 8cc702081a6439ec
90befffa23631e28 a4506cebde82bde9 bef9a3f7b2c67915 c67178f2e372532b
ca273eceea26619c d186b8c721c0c207 eada7dd6cde0eb1e f57d4f7fee6ed178
06f067aa72176fba 0a637dc5a2c898a6 113f9804bef90dae 1b710b35131c471b
28db77f523047d84 32caab7b40c72493 3c9ebe0a15c9bebc 431d67c49c100d4c
4cc5d4becb3e42b6 597f299cfc657e2a 5fcb6fab3ad6faec 6c44198c4a475817.
```

Gambar 3 konstan SHA – 512

Initialization Vector

a = 0x6A09E667F3BCC908	b = 0xBB67AE8584CAA73B
c = 0x3C6EF372FE94F82B	d = 0xA54FF53A5F1D36F1
e = 0x510E527FADE682D1	f = 0x9B05688C2B3E6C1F
g = 0x1F83D9ABFB41BD6B	h = 0x5BE0CD19137E2179

Gambar 4 *first buffers* SHA - 512

4. Perhitungan buffer dengan 80 round perhitungan dengan menggunakan persamaan 2.

$$T_1 = h + \sum_1 (e) + Ch(e,f,g) + K_j + W_j \quad (2)$$

$$T_2 = \sum_0 (a) + Maj(a,b,c)$$

$$h = g$$
$$g = f$$
$$f = e$$
$$e = d + T_1$$
$$d = c$$
$$c = b$$
$$b = a$$
$$a = T_1 + T_2$$

Keterangan :
Conditional Function

$$Ch(e,f,g) = (e_j \; AND \; f_j) \oplus (NOT \; e_j \; AND \; g_j)$$

Majority Function

$$Maj(a,b,c) = (a_j \; AND \; b_j) \oplus (b_j \; AND \; c_j) \oplus (c_j \; AND \; a_j)$$

Rotate Function

$$\sum_0 (a) = ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x)$$

$$\sum_1 (a) = ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x)$$

$\oplus$ = XOR
$ROTR^n(x)$ = Circular Right Shift (Rotasi) dari argumen x dengan n bits

5. Setelah mendapatkan hasil dari perhitungan terakhir maka final buffers akan ditambahkan dengan buffers sebelumnya dengan menggunakan persamaan 3.

$$a = a_0 + a_{79}$$
$$b = b_0 + b_{79}$$
$$c = c_0 + c_{79}$$
$$d = d_0 + d_{79}$$
$$e = e_0 + e_{79}$$
$$f = f_0 + f_{79}$$
$$g = g_0 + g_{79}$$
$$h = h_0 + h_{79}$$

proses perhitungan diatas akan dilakukan kembali untuk blok selanjutnya hingga blok terakhir. Setelah proses perhitungan ini sudah selesai pada blok terakhir, maka hasil hash code yang terakhir adalah hash code final yang akan menjadi output pada fungsi hash SHA – 512.

2.4 Flutter

Flutter merupakan framework baru yang dibuat oleh google untuk membuat aplikasi mobile baik di android, ios hingga web dan juga desktop. Aplikasi yang dibuat dengan flutter ditulis dengan bahasa pemrograman dart yang memiliki fitur yang sama dengan bahasa pemrograman modern lainnya seperti kotlin dan swift. Penggunaan bahasa pemrograman dart membuat flutter mempunyai beberapa kelebihan dibandingkan dengan yang lainya seperti *hot reload* yang berfungsi ketika adanya perubahan pada program, maka dapat dilihat secara langsung karena hanya membutuhkan beberapa detik untuk melakukan proses build dari program. Dan fitur yang menjadi kelebihan flutter yaitu *cross platform* yang dapat memudahkan ketika ingin membangun aplikasi pada 2 platform (android & ios) hanya dengan satu *codebase* saja. [4]

2.5 REST API

API adalah singkatan dari application programming interface yaitu sebuah software yang memungkinkan para developer untuk mengintegrasikan dan mengizinkan dua aplikasi yang berbeda secara bersamaan untuk saling terhubung dengan satu sama lain. Salah satu jenis dari API adalah REST API yang mempunyai desain arsitektur data yang akan diberikan oleh server ke aplikasi dengan berupa format text, JSON, atau XML.[5]

Adapun metode HTTP yang secara umum dipakai dalam REST API yaitu :

1. GET, berfungsi untuk membaca data dari server.
2. POST, berfungsi untuk membuat atau menambahkan data baru di server.
3. PUT, berfungsi untuk memperbaharui data yang terdapat pada server.
4. DELETE, berfungsi untuk menghapus data dari server.
5. OPTIONS, mendapatkan operasi yang didukung pada *resource* dari server.

3. Hasil Pengujian

3.1 Pengujian *QR Code*

Pengujian *QR Code* ini bertujuan untuk mengetahui sejauh mana *QR Code* dapat bekerja dan dapat digunakan. Pada pengujian ini akan menggunakan angka 1 dan 0 untuk mengindikasi keberhasilan atau kegagalan pada penelitian yang dilakukan dimana 1 = berhasil dan 0 = gagal.

1. Pengujian Kondisi *QR Code*
   Pengujian kondisi ini dilakukan dengan beberapa kondisi untuk melihat apakah kondisi dari *QR Code* akan mempengaruhi proses pembacaan kontennya. Hasil pengujian ini dapat dilihat pada tabel 3. Dibawah ini dijelaskan untuk hasil pengujian kondisi maka *QR Code* tidak dapat dibaca jika kerusakan atau kondisi dari *QR Code* diatas 100%.

Tabel 3 Hasil Pengujian Kondisi *QR Code*

No.	Deskripsi	Kondisi *QR Code*	Proses Pemindai
1	Menguji pembacaan *QR Code* dengan baik dan benar	100 %	1
2	Menguji pembacaan *QR Code* yang tertutup sedikit pada bagian pojok bawah kanan	90 %	1
3	Menguji pembacaan *QR Code* yang tertutup sedikit pada bagian pojok bawah kiri	90 %	1
4	Menguji pembacaan *QR Code* yang tertutup sedikit pada bagian pojok atas kanan	90 %	1
5	Menguji pembacaan *QR Code* yang tertutup sedikit pada bagian pojok atas kiri	90 %	1
6	Menguji pembacaan *QR Code* yang tertutup setengah	50 %	0
7	Menguji pembacaan *QR Code* yang terbuka sedikit	20 %	0
8	Menguji pembacaan *QR Code* yang salah satu findernya tertutup	70 %	0
9	Menguji pembacaan *QR Code* yang dua findernya tertutup	40%	0

2. Pengujian Jarak *QR Code*
   Pengujian jarak ini dilakukan dengan menguji pembacaan *QR Code* dengan jarak yang berbeda-beda. Untuk jarak yang akan dipakai yaitu di bawah 30 cm dan diatas 50 cm. Untuk hasil pengujian jarak dari *QR Code* dapat dilihat pada tabel 4. Tabel dibawah ini menjelaskan jika pemindaian dapat dilakukan jika kamera sudah dalam radius dibawah 50 cm.

Tabel 4 Hasil Pengujian Jarak *QR Code*

No.	Deskripsi	Proses Pemindai
1	Menguji pembacaan *QR Code* dengan jarak < 0.3 meter	1
2	Menguji pembacaan *QR Code* dengan jarak > 0.5 meter	0

3.2 Pengujian Hasil *Scanning QR Code*

Pengujian hasil scanning merupakan pengujian yang berfokus pada kebutuhan fungsional dari aplikasi. Pengujian ini dilakukan untuk melihat hasil keluaran yang diharapkan dari sistem yang diuji, apakah dapat berjalan sesuai yang diharapkan atau tidak. Untuk hasil dari pengujian scanning *QR Code* dapat dilihat pada tabel dibawah ini.

Tabel 5 Hasil Pengujian Scanning *QR Code*

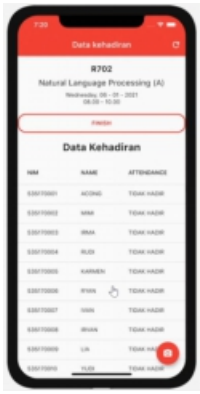
No	Skenario Pengujian	Hasil Yang Diharapkan	Hasil Pengujian
1	*Scan QR Code* mahasiswa yang terdaftar dengan aplikasi smart presensi	Menampilkan data mahasiswa	Berhasil
2	*Scan QR Code* mahasiswa dengan aplikasi *QR Code Scanner*	Menampilkan *Hash Code*	Berhasil
3	*Scan QR Code* yang terdapat pada tiket parkir	Menampilkan error bahwa kode tidak dikenali	Berhasil
4	*Scan QR Code* mahasiswa yang tidak terdaftar dalam ujian yang sedang diawasi pengawas	Menampilkan error bahwa kode tidak dikenali	Berhasil

3.3 Pengujian *Mobile Application*

Pengujian mobile application merupakan pengujian untuk program mobile yang telah dibuat menggunakan flutter dan melihat berhasilnya menjalankan aplikasi smart presensi ini di dalam 2 platform yaitu android dan ios. Untuk hasil dari pengujian dapat dilihat pada tabel 6.

Tabel 6 pengujian *mobile application*

Android		
Device	Screenshot	Running

Xiaomi Note 8		Success
iOS		
Device	Screenshot	Running
Emulator Iphone XS Max		Success

4. Kesimpulan

Berdasarkan pengujian program "*Smart Presensi* Menggunakan *QR Code* Dengan Secure Hash Algorithm 2 (SHA – 2) ", maka didapatkan kesimpulan sebagai berikut :

1. Pada pengujian scanning *QR Code* mendapatkan isi dari *QR Code* yang dimiliki setiap mahasiswa akan berupa hasil fungsi hash SHA – 512 dari data mahasiswa, dan hal ini akan menjadikan setiap data mahasiswa akan terlindungi karena data mahasiswa hanya akan menampilkan datanya jika *QR Code* mahasiswa tersebut dipindai oleh aplikasi mobile yang dimiliki oleh pengawas ujian.
2. Pada pengujian *mobile application* mendapatkan kode program aplikasi mobile yang dibuat dengan menggunakan *framework* flutter dapat dijalankan pada operasi sistem android dan ios dengan hanya menggunakan satu *code base*.
3. Penggunaan fungsi hash SHA – 512 akan menghasilkan output yang mempunyai *digest message* berukuran 512 bit dan hal inilah yang menjadikan fungsi hash ini menjadi yang paling aman karena sulit untuk dipecahkan di bandingkan dengan semua jenis arsitektur dari keluarga fungsi hash SHA – 2.
4. Pengujian *Scanning QR Code* dapat disimpulkan bahwa scanner tidak dapat melakukan *scanning* pada *QR Code* lain selain yang dimiliki mahasiswa untuk ujian serta *QR Code* tidak dapat diganti dengan yang lainnya.
5. Pengujian *QR Code* dapat disimpulkan bahwa *QR Code* dapat berfungsi dengan baik jika tidak terjadi kerusakan pada *QR Code* diatas 30%.

REFERENSI

[1] Denso Wave Incorporated, What is a QR Code? , qrcode.com , 18 September 2020.

[2] MD Shariful Islam, https://www.slideshare.net/sharifulr/secure-hash-algorithm-sha512, 21 September 2020.

[3] Zaid Khaishagi, Cryptography:Examplaining SHA 512, https://medium.com/@zaid960928/cryptography-explaining-sha-512-ad896365a0c1, 11 September 2020

[4] Utrodus Said Al Baqi, Pengenalan Flutter & Dart, https://medium.com/@sekolahflutter/pengenalan-dart-flutter-26a056638bc9 , 15 September 2020.

[5] Beon Intermedia, Apa itu API? Mengenal Lebih Jauh tentang Web API dan Web Service, https://www.jagoanhosting.com/blog/apa-itu-web-api/ , 26 September 2020.

Ivan Wijaya, mahasiswa pada program studi Fakultas Teknologi Informasi di Universitas Tarumanagara.