

# Implementasi Algoritma Reed Solomon Codes Pada Proses Encoding QR Code pada Sistem Absensi

Andez Apriansyah<sup>1</sup>, Fauziah<sup>2</sup>, Nur Hayati<sup>3</sup>

<sup>1,2,3</sup> Jurusan Informatika Universitas Nasional

<sup>1</sup> andezapriansyah@gmail.com, <sup>2</sup> fauziah@civitas.unas.ac.id, <sup>3</sup> nurhayati@civitas.unas.ac.id

**Abstrak**— Sistem absensi yang dilakukan saat ini masih banyak yang menggunakan sistem secara manual, antara lain dengan tanda tangan di media kertas. Dengan sistem seperti ini tentunya terdapat kekurangan dari segi efisiensi, mulai dari efisien waktu, tenaga, dan biaya. Oleh karena itu, penelitian ini bertujuan untuk menjadi proses absensi menjadi lebih efisien, mengganti sistem absensi menggunakan teknologi QR code, menentukan jenis QR code untuk absensi berdasarkan pengujian dengan tingkat koreksi kesalahan dan dengan kondisi yang berbeda. Metode yang diterapkan dalam penelitian ini menggunakan Reed Solomon sebagai koreksi kesalahan pada proses encoding QR code. Penelitian ini dibuktikan melalui serangkaian pengujian dengan jumlah data sebanyak 100 data dan total pengujian sebanyak 4000 kali meliputi tingkat koreksi kesalahan, jarak scan, dan kondisi kartu atau kertas. Hasil dari pengujian tersebut berupa scan yang terbaca pada kondisi baik dengan tingkat koreksi kesalahan L = 63.4%, M = 73%, Q = 79%, H = 92% sedangkan di kondisi rusak L = 41.8% , M = 57.6%, Q = 65.4%, H = 80.8%. Oleh karena itu, tingkat koreksi kesalahan ini mempengaruhi QR Code.

**Kata kunci**— Absensi, biner, encoding, qr code, reed solomon.

**Abstract**— The attendance system that is done at this time there is still many who use the system manually, among others, with signatures on paper media. With a system like this, of course, there are deficiencies in terms of efficiency, starting from the efficient time, effort and cost. Therefore, this study aims to make the attendance process more efficient, replace the attendance system using QR code technology, determine the type of QR code for attendance based on testing with error correction levels and with different conditions. The method applied in this study uses Reed Solomon as error correction in the QR Code encoding process. This research is proven through a series of tests with 100 data and total testing 4000 times including error correction rate, scan distance, and card or paper condition. The results of the test are scans that are read in good condition with a correction level L = 63.4%, M = 73%, Q = 79%, H = 92% while in damaged conditions L = 41.8%, M = 57.6%, Q = 65.4%, H = 80.8%. Therefore, this level of error correction affects the QR Code.

**Keywords**— Attendance, biner, encode, qr code, reed solomon.

## I. PENDAHULUAN

Sistem absensi yang dilakukan saat ini masih banyak yang menggunakan sistem secara manual, antara lain dengan tanda tangan di media kertas. Dengan sistem seperti ini tentunya terdapat kekurangan dari segi efisiensi, mulai dari efisien waktu, tenaga, dan biaya.

QR code merupakan sebuah teknologi yang dapat diterapkan sebagai pengganti sistem absensi manual [1]. Dengan teknologi ini sistem absensi dapat menjadi lebih efisien.

Dalam penelitian sebelumnya yang dilakukan oleh Qurotul Aini dkk. Dengan judul “Penerapan QR Code Sebagai Media Pelayanan Untuk Absensi Pada Website Berbasis PHP Native” dilakukan menggunakan metode md5 dalam menjelaskan proses encoding [2]. Penelitian selanjutnya dengan judul “Smart Presensi Menggunakan Qr-Code Dengan Enkripsi Vigenere Cipher” yang dilakukan oleh Moh. Lukman Sholeh melakukan penelitian tersebut dengan menerapkan metode Vigenere Chiper sebagai encoding-nya [3]. Kemudian di penelitian lain oleh Edwin Fajar Nurdiansyah berjudul “Implementasi Qrcode Sebagai Tiket Masuk Event Dengan Memperhitungkan Tingkat Koreksi Kesalahan” dengan menerapkan metode Reed Solomon pada encoding-nya yang berguna sebagai koreksi kesalahan [4].

Berdasarkan penelitian-penelitian tersebut, maka metode Reed Solomon memiliki kemampuan mengkoreksi kesalahan pada proses encoding QR Code. QR code dibuat dengan versi 2D, karena telah dikodekan secara vertical dan arah horizontal, desain 2D tersebut digunakan sebagai media penyimpanan data [5]-[6]. Penelitian ini menitik beratkan pada proses encoding dengan memperhitungkan koreksi kesalahan dalam QR code. Dengan tujuan untuk mewujudkan paperless dalam sistem d absensi, menentukan tingkat koreksi kesalahan yang baik berdasarkan kondisi QR code, dan menentukan jenis QR code

yang tepat untuk digunakan pada sistem absensi berdasarkan tingkat koreksi kesalahan.

## II. METODOLOGI PENELITIAN

Metode yang diterapkan dalam penelitian ini menggunakan Reed Solomon [7] sebagai koreksi kesalahan pada proses encoding QR code [8]. Reed Solomon yang dirumuskan Irving Reed dan Gus Solomon menggambarkan secara sistematis sebuah codewords (unit yang membangun area data) yang mampu mengoreksi kesalahan secara acak pada data yang diterima berdasarkan aritmatik Galois Field (GF) [4].

TABEL I  
TINGKAT KOREKSI KESALAHAN

Tingkat Koreksi Kesalahan	Lambang Tingkat Koreksi Kesalahan	Kemampuan Koreksi Kesalahan
Low	L	7%
Medium	M	15%
Quartile	Q	25%
High	H	30%

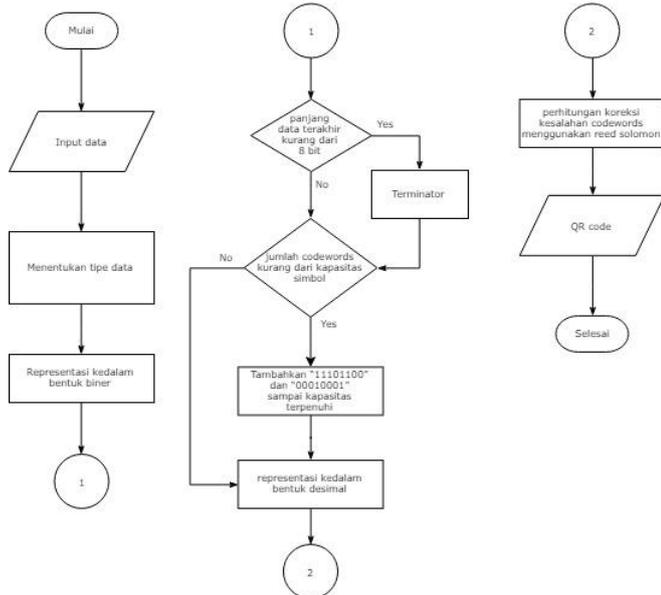
(Sumber: qrcode.com)

Untuk melakukan koreksi kesalahan pada QR code metode Reed Solomon membagi menjadi beberapa tingkat koreksi kesalahan [9]-[10], seperti pada Tabel I.

QR code menerapkan koreksi kesalahan untuk menghasilkan serangkaian codewords yang ditambahkan pada urutan codewords yang memungkinkan simbol untuk dibaca.

Tahapan dari rancangan sistem yang dibangun dibagi menjadi dua, yaitu:

A. Analisis Encoding



Gambar 1. Flowchart Encoding

Gambar 1 menggambarkan tahap perancangan aplikasi yaitu merubah data menjadi QR code (encoding) [11].

TABEL II  
TIPE INPUT

Type Input	Bit/Karakter	Type Karakter	Indikator
Numeric	10/3	0-9	0001
Alphanumerik	11/2	0-9, A-Z, \$, %, *, +, -, ., /, :	0010
Byte	8	ISO 8859-1	0100
Kanji	13	Shift JIS X 0208	1000

1. Input data, sebagai contoh adalah “UNAS1949” yang bertipe *alphanumeric* berdasarkan pada Tabel II.
2. Indikator yang dipilih 0010. Pada versi 1-9, Tipe input berupa *numeric* memiliki panjang 10bit, sedangkan untuk *alphanumeric* 9bit. Contoh input yang dipakai memiliki 8 karakter bertipe *alphanumeric*, maka 8 karakter dirubah menjadi biner sepanjang 9 bit hasilnya 000001000.

TABEL III  
KODE ALPHANUMERIC

Kode	Karakter	Kode	Karakter	Kode	Karakter
0	0	16	G	32	W
1	1	17	H	33	X
2	2	18	I	34	Y
3	3	19	J	35	Z
4	4	20	K	36	[sp]
5	5	21	L	37	\$
6	6	22	M	38	%
7	7	23	N	39	*
8	8	24	O	40	+
9	9	25	P	41	-
10	A	26	Q	42	.
11	B	27	R	43	/
12	C	28	S	44	:
13	D	29	T		
14	E	30	U		
15	F	31	V		

3. Data tersebut direpresentasikan kedalam biner, kode biner dari karakter *alphanumeric* dapat dilihat pada Tabel III. Data inputan dikelompokkan per 2 karakter, lalu karakter

pertama dikalikan 45 kemudian hasilnya ditambah karakter ke-2.

“UN”	“AS”	“19”	“49”
45*30+23	45*10+28	45*1+9	45*4+9
1373	478	54	189
10101011101	00111011110	00000110110	00010111101

4. Tambahkan biner 0000 sebagai terminator dalam representasi biner Tabel II. Ketika panjang data yang di *encoding* penuh dalam versi dan tingkat koreksi kesalahan, maka terminator tidak diperlukan. Kemudian representasi biner tersebut di gabungan per 8 bit, sehingga menjadi

**00100000 01000101 01011101 00111011 11000000  
11011000 01011110 10000000**

Jika panjang data terakhir kurang dari 8 bit, maka ditambah 0, sehingga

**00100000 01000101 01011101 00111011 11000000  
11011000 01011110 10000000**

5. Jika perhitungan *codewords* kurang dari kapasitas simbol yaitu 9 dalam versi dan tingkat koreksi kesalahan, maka ditambahkan “11101100” dan “00010001” sampai kapasitas terpenuhi. Sehingga

**00100000 01000101 01011101 00111011 11000000  
11011000 01011110 10000000 11101100**

6. Biner tersebut dirubah menjadi desimal

**32 69 93 59 192 216 94 128 236**

Kemudian melakukan koreksi kesalahan *codewords* menggunakan Reed Solomon. Pertama, hasil tersebut dibatasi sesuai ketentuan dari RS blok. Dalam hal ini RS Blok yang digunakan versi 1 tingkat koreksi kesalahannya H, maka jumlah koreksi kesalahan *codewords*-nya yaitu 17. Selanjutnya pilih fungsi  $g(x)$  dari 17, yaitu

$$g(x) = x^{17} + a^{43}x^{16} + a^{139}x^{15} + a^{206}x^{14} + a^{78}x^{13} + a^{43}x^{12} + a^{239}x^{11} + a^{123}x^{10} + a^{206}x^9 + a^{214}x^8 + a^{147}x^7 + a^{24}x^6 + a^{99}x^5 + a^{150}x^4 + a^{39}x^3 + a^{243}x^2 + a^{163}x + a^{136} \quad (1)$$

Fungsi tersebut adalah bentuk dasar dari elemen  $GF(2^8)$ . Kemudian mempersiapkan polinomial  $f(x)$ , yaitu

$$f(x) = 32x^{25} + 69x^{24} + 93x^{23} + 59x^{22} + 192x^{21} + 216x^{20} + 94x^{19} + 128x^{18} + 236x^{17} \quad (2)$$

Koefisien istilah terkemuka dalam  $f(x)$  adalah 32. Untuk 32 adalah  $\alpha^5$ , maka digunakan fungsi

$$g(x)a^5x^8 \quad (3)$$

$$= a^5x^{25} + a^5a^{43}x^{24} + a^5a^{139}x^{23} + a^5a^{206}x^{22} + a^5a^{78}x^{21} + a^5a^{43}x^{20} + a^5a^{239}x^{19} + a^5a^{123}x^{18} + a^5a^{206}x^{17} + a^5a^{214}x^{16} + a^5a^{147}x^{15} + a^5a^{24}x^{14} + a^5a^{99}x^{13} + a^5a^{150}x^{12} + a^5a^{39}x^{11} + a^5a^{243}x^{10} + a^5a^{163}x^9 + a^5a^{136}x^8 \quad (4)$$

$$= a^5x^{25} + a^{48}x^{24} + a^{144}x^{23} + a^{211}x^{22} + a^{83}x^{21} + a^{48}x^{20} + a^{244}x^{19} + a^{128}x^{18} + a^{211}x^{17} + a^{219}x^{16} + a^{152}x^{15} + a^{29}x^{14} + a^{104}x^{13} + a^{155}x^{12} + a^{44}x^{11} + a^{248}x^{10} + a^{168}x^9 + a^{141}x^8$$

$$= 32x^{25} + 70x^{24} + 168x^{23} + 178x^{22} + 187x^{21} + 70x^{20} + 250x^{19} + 133x^{18} + 178x^{17} + 86x^{16} + 73x^{15} + 48x^{14} + 13x^{13} + 114x^{12} + 238x^{11} + 27x^{10} + 252x^9 + 21x^8$$

Kemudian lakukan operasi XOR hasil dari (2) dengan (4)

$$= (32\oplus 32)x^{25} + (69\oplus 70)x^{24} + (93\oplus 168)x^{23} + (59\oplus 178)x^{22} + (192\oplus 187)x^{21} + (216\oplus 70)x^{20} + (94\oplus 250)x^{19} + (128\oplus 133)x^{18} + (236\oplus 178)x^{17} + (0\oplus 86)x^{16} + (0\oplus 73)x^{15} + (0\oplus 48)x^{14} + (0\oplus 13)x^{13} + (0\oplus 114)x^{12} + (0\oplus 238)x^{11} + (0\oplus 27)x^{10} + (0\oplus 252)x^9 + (0\oplus 21)x^8$$

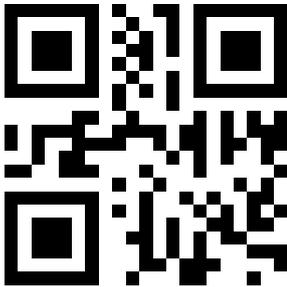
Maka didapatkan hasil sebagai berikut

$$3x^{24} + 245x^{23} + 137x^{22} + 123x^{21} + 158x^{20} + 164x^{19} + 5x^{18} + 94x^{17} + 86x^{16} + 73x^{15} + 48x^{14} + 13x^{13} + 114x^{12} + 238x^{11} + 27x^{10} + 252x^9 + 21x^8$$

Dan desimal yang dihasilkan dari hasil operasi XOR adalah sebagai berikut

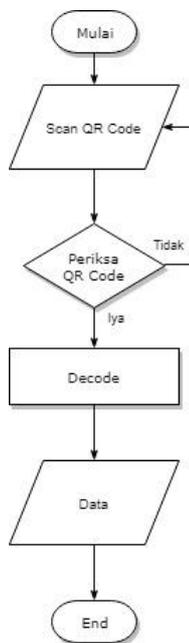
**3 245 137 123 158 164 5 94 86 73 48 13 114 238 27 252 21**

7. Dari data tersebut dapat dihasilkan QR code seperti Gambar 2.



Gambar 2. QR code

### B. Analisis Decoding



Gambar 3. Flowchart Decoding

Kemudian untuk melakukan proses absensi dilakukan proses *decoding* seperti Gambar 3, yaitu perubahan QR code menjadi data.

1. Dari hasil *encoding* yaitu QR code di scan dengan menggunakan kamera *webcam*.
2. Memeriksa apakah QR Code tersebut telah terdaftar atau tidak, jika tidak maka proses akan kembali ke scan. Jika terdaftar maka akan masuk proses *decoding*
3. Proses *decoding* ini perubahan QR code menjadi data dengan *library* yang ada
4. Setelah itu proses absen berhasil

## III. HASIL DAN PEMBAHASAN

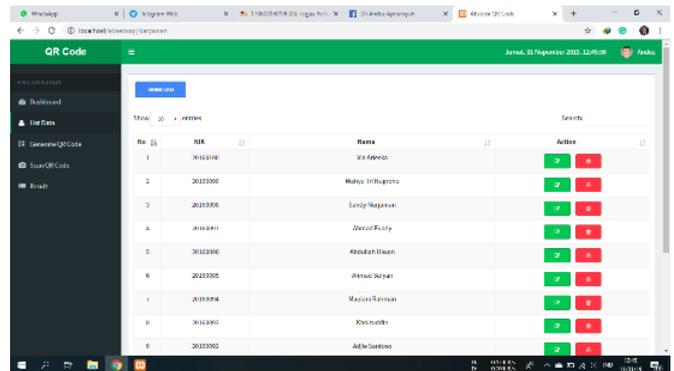
### A. Analisis Kebutuhan

Pada proses pembuatan aplikasi absensi dalam penelitian ini dibutuhkan beberapa perangkat pendukung seperti perangkat lunak (*software*) dan perangkat keras (*hardware*), yang diantaranya:

1. Kebutuhan *Hardware* : *Webcam* dari Laptop Intel Core i3-6006U, 2.0GHz dengan RAM 12GB dan HDD 500GB
2. Kebutuhan *Software* : Visual Studio Code, XAMPP, Google Chrome, Draw.io, Ms. Excel.

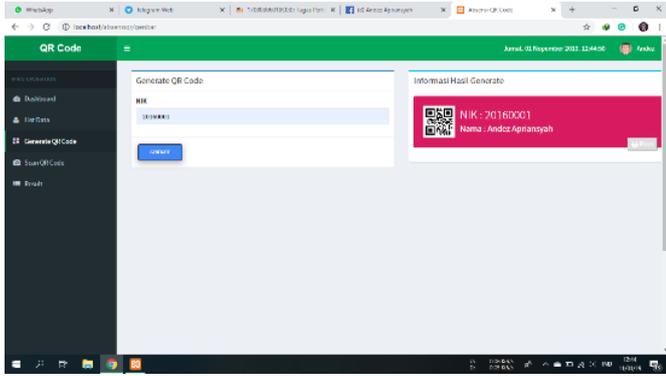
### B. Tampilan Aplikasi Absensi

Aplikasi yang dirancang sebagai sistem absensi dalam penelitian ini berbasis website. Website ini sebagai aplikasi sistem *encoding* serta *decoding* QR code. Untuk melakukan proses *encoding* diperlukan data-data terlebih dahulu.



Gambar 4. Tampilan List Data

Pada Gambar 4 terdapat beberapa *button* yaitu *button* tambah data untuk menambah data, *button* edit yang digunakan untuk merubah data dan *button* delete yang digunakan untuk menghapus data.



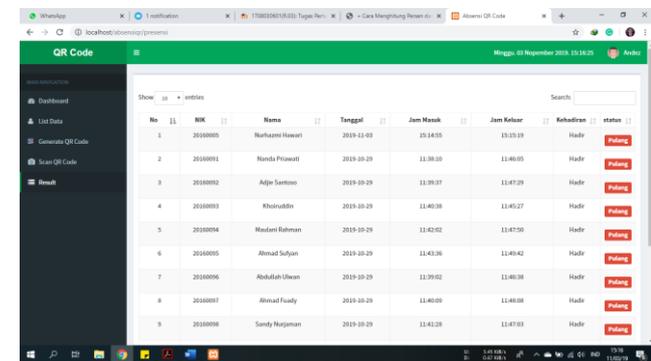
Gambar 5. Tampilan Generate QR Code

Gambar 5 merupakan proses *generate QR code*, untuk melakukan proses *generate QR code* dalam aplikasi tersebut dengan menginput NIK yang sudah terdaftar atau telah di tambahkan pada proses tambah data pada Gambar 4. Hasil dari *generate QR code* tersebut akan muncul didalam form “Informasi Hasil Generate” di samping form “Generate QR Code”. Kemudian print hasil citra tersebut, dan *scan QR code* tersebut seperti pada Gambar 6.



Gambar 6. Kamera Scan QR Code

Setelah *scan* citra QR Code berhasil, maka data kehadiran akan otomatis terinput dan ditampilkan dalam menu *result* pada aplikasi absensi seperti Gambar 7.



Gambar 7. Tampilan Result Absensi

Gambar 7 merupakan hasil atau *result* dari proses *scan QR code*, *result* ini akan menampilkan data awal yaitu berupa NIK dan nama yang telah diinput oleh *user* pada proses input data, kemudian di *result* juga akan memunculkan jam serta keterangan masuk maupun keluar.

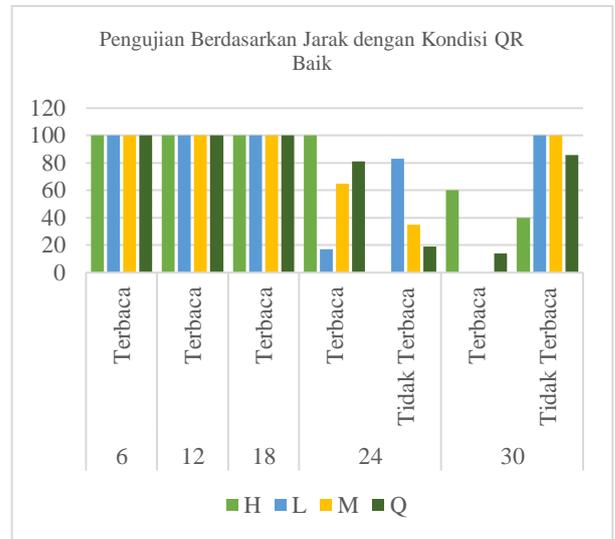
### C. Pengujian

Dalam proses pengujian aplikasi ini, jumlah data yang diuji pada penelitian ini sebanyak 100 data dengan total pengujian sebanyak 4000 kali pengujian. Pengujian tersebut meliputi tingkat koreksi kesalahan, jarak *scan QR code*, dan kondisi kartu atau kertas.

TABEL IV  
KONDISI QR CODE

QR Code	Kondisi	Keterangan
	Baik	Faktor penyebab tidak terdeteksi dikarenakan pencahayaan yang kurang maksimal
	Rusak karena coretan	Penyebab tidak terdeteksi dikarenakan coretan yang tebal
	Rusak karena basah	Penyebab tidak terdeteksi dikarenakan basah

Tabel IV merupakan kondisi kartu atau kertas yang dibagi kedalam 2 kondisi, yaitu baik dan rusak. Untuk kondisi kartu yang baik yaitu citra *QR code* yang bersih tanpa adanya coretan, kotor maupun rusak, sedangkan untuk kondisi kartu yang rusak yaitu ditandai dengan adanya coretan didalam citra *QR code* tersebut.



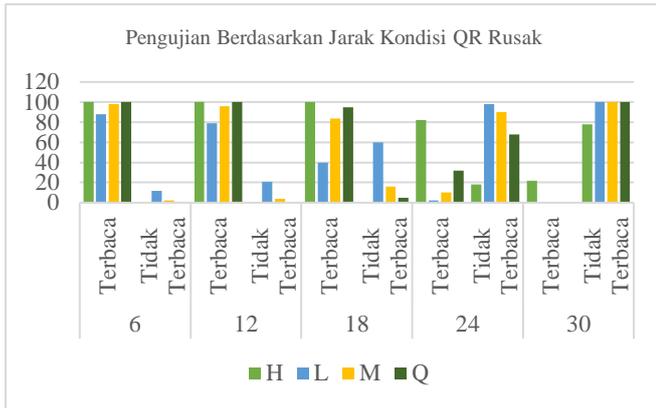
Gambar 8. Pengujian berdasarkan jarak dengan kondisi baik

Pada gambar 8 merupakan grafik yang menunjukkan pengujian berdasarkan jarak *scan QR code* dengan kondisi kartu atau kertas *QR code* dalam keadaan baik. Dengan Y axis yaitu *range* 0 - 120 yang merupakan jumlah data yang telah diinput kedalam database aplikasi. Kemudian pada X axis yaitu *range* jarak 6 cm, 12 cm, 18 cm, 24 cm, 30 cm yang menggunakan kelipatan 6 digunakan untuk mengukur jarak *scan QR code*.

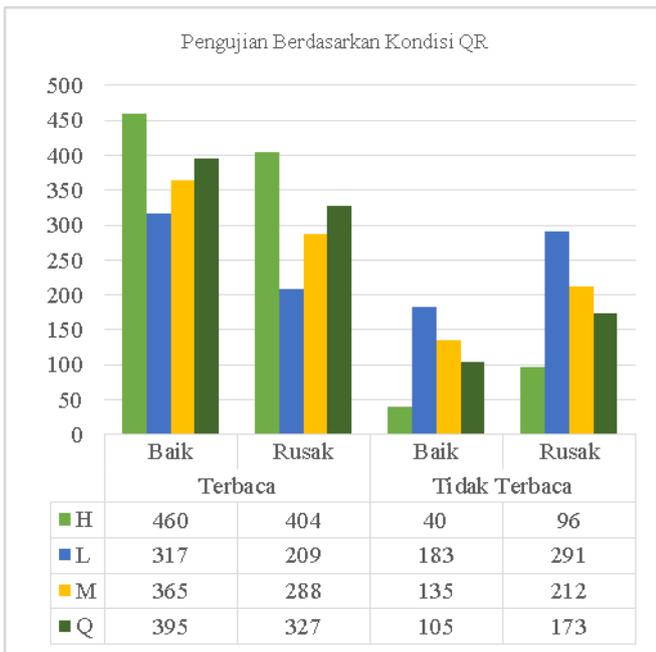
Untuk kondisi kartu atau kertas *QR code* dalam keadaan baik di jarak 6 cm, 12 cm, dan 18 cm dengan semua tingkat koreksi kesalahan dari *low* (L) sampai *high* (H), data yang terbaca sebanyak 100 data. Untuk jarak 24cm, data yang terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 17 data, *medium* (M) = 65 data, *quartile* (Q) = 81 data, *high* (H) = 100 data dan untuk data yang tidak terbaca pada jarak 24 cm yaitu *low* (L) = 83 data, *medium* (M) = 35 data, *quartile* (Q) = 19 data, *high* (H) = 0 data. Sedangkan di jarak 30 cm, data yang terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 0 data, *medium* (M) = 0 data, *quartile* (Q) = 14, *high*

(H) = 60 data dan untuk data yang tidak terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 100 data, *medium* (M) = 100 data, *quartile* (Q) = 86 data, *high* (H) = 40 data.

data, dan data yang tidak terbaca yaitu *low* (L) = 98 data, *medium* (M) = 90 data, *quartile* (Q) = 68 data, *high* (H) = 18 data. Sedangkan di jarak 30 cm, data yang terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 79 data, *medium* (M) = 0 data, *quartile* (Q) = 0 data, *high* (H) = 22 data, dan data yang tidak terbaca yaitu *low* (L) = 100 data, *medium* (M) = 100 data, *quartile* (Q) = 100 data, *high* (H) = 78 data.



Gambar 9. Pengujian berdasarkan jarak dengan kondisi rusak



Gambar 10. Pengujian berdasarkan kondisi

Dalam kondisi QR code rusak di jarak 6 cm, data yang terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 88 data, *medium* (M) = 98 data, *quartile* (Q) = 100 data, *high* (H) = 100 data dan untuk data yang tidak terbaca yaitu *low* (L) = 12 data, *medium* (M) = 2 data, *quartile* (Q) = 0 data, *high* (H) = 0 data. Sedangkan di jarak 12 cm, data yang terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 79 data, *medium* (M) = 96 data, *quartile* (Q) = 100 data, *high* (H) = 100 data dan data yang tidak terbaca yaitu *low* (L) = 21 data, *medium* (M) = 4 data, *quartile* (Q) = 0 data, *high* (H) = 0 data. Sedangkan di jarak 18 cm, data yang terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 40 data, *medium* (M) = 84 data, *quartile* (Q) = 95, *high* (H) = 100 data, dan data yang tidak terbaca yaitu *low* (L) = 60 data, *medium* (M) = 16 data, *quartile* (Q) = 5 data, *high* (H) = 0 data. Sedangkan di jarak 24 cm, data yang terbaca pada masing-masing tingkat koreksi kesalahan yaitu *low* (L) = 2 data, *medium* (M) = 10 data, *quartile* (Q) = 32 data, *high* (H) = 82

Dari grafik pada Gambar 10 untuk mendapatkan sebuah persentase dari hasil pengujian tersebut didapatkan dengan rumus

$$p = \frac{n}{m} \times 100 \quad (5)$$

Dimana *p* adalah hasil persentase, *n* adalah nilai, dan *m* adalah total nilai keseluruhan.

TABEL V  
PERSENTASE PENGUJIAN

Tingkat Koreksi Kesalahan	Baik		Rusak	
	Terbaca (%)	Tidak Terbaca (%)	Terbaca (%)	Tidak Terbaca (%)
L	63.4	36.6	41.8	58.2
M	73	27	57.6	42.4
Q	79	21	65.4	34.6
H	92	8	80.8	19.2

Maka hasil persentase data yang terbaca dan tidak terbaca dengan kondisi baik rusak didapat hasil presentase seperti pada Tabel V.

Scan yang terbaca pada kondisi baik dengan tingkat koreksi *low* (L) = 63.4%, *medium* (M) = 73%, *quartile* (Q) = 79%, *high* (H) = 92%. Sedangkan dalam kondisi kartu atau kertas QR code rusak yaitu *low* (L) = 41.8%, *medium* (M) = 57.6%, *quartile* (Q) = 65.4%, *high* (H) = 80.8%. Oleh karena itu, tingkat koreksi kesalahan ini dapat mempengaruhi hasil dari proses decoding QR code.

Penggunaan QR code pada absensi di pabrik disarankan menggunakan tingkat *Quartile* atau *High* yang memiliki presentase 80.8% pada kondisi rusak, karena area pabrik yang dapat menyebabkan QR code rentan rusak atau kotor. Sedangkan untuk di area pendidikan ataupun kantor bisa menggunakan *Low*, karena di area tersebut QR Code tidak cepat kotor atau rusak. Akan tetapi lebih baik lagi di tingkat *Medium* keatas, karena persentase *Medium* di kondisi rusak 57.6% terbaca.

#### IV. KESIMPULAN

Berdasarkan hasil pembahasan diatas, dengan menerapkan sistem absensi menggunakan QR code, maka dapat mewujudkan *paperless* dalam proses absensi sehingga menjadikan sistem absensi menjadi lebih efisien. Tingkat koreksi kesalahan QR code yang baik adalah dengan menggunakan tingkat *high* (H), karena pada tingkat tersebut persentase terbaca pada kondisi baik adalah 92%, sedangkan pada kondisi rusak mencapai 80.8%. Dan untuk penggunaan QR code dalam sistem absensi bergantung kondisi lingkungan sekitar absensi. Jika kondisi lingkungan bersih maka tingkat koreksi kesalahan yang digunakan bisa dengan tingkat *Low* (L) keatas, karena ditingkat *Low* (L) persentase terbaca pada kondisi QR code baik mencapai 63.4%. akan tetapi lebih disarankan untuk menggunakan tingkat *medium* (M) agar proses lebih decoding lebih baik karena persentase terbaca

dikondisi baik mencapai 73%. Jika lingkungan sekitar rentan kotor atau rusak maka tingkat koreksi yang digunakan *Quartile* (Q) keatas, karena ditingkat *Quartile* persentase terbaca pada kondisi QR code rusak mencapai 65.4%.

#### REFERENSI

- [1] S. Setyorini and J. Arifin, "Pemanfaatan Qr Code Untuk Perekaman Data Kehadiran Siswa Terintegrasi Dengan Sistem Informasi Manajemen Sekolah Smk Mahardika Malang," *Netw. Eng. Res. Oper.*, vol. 4, no. 1, pp. 5–13, 2018.
- [2] Q. Aini, U. Rahardja, and A. Fatillah, "Penerapan QRCode Sebagai Media Pelayanan Untuk Absensi Pada Website Berbasis PHP Native Application of QRCode as Media Services for Attendance on the Website Based Php Natively," *J. Ilm. Sist. Inf. dan Tek. Inform.*, vol. 8, no. 1, pp. 47–56, 2018.
- [3] M. L. Sholeh and L. A. Muharom, "Smart Presensi Menggunakan QR-Code Dengan Enkripsi Vigenere Cipher," *Limits (Journal Math. Its Appl.*, vol. 13, no. 2, pp. 31–44, 2016.
- [4] E. F. Nurdiansyah and I. Afrianto, "Implementasi QR Code Sebagai Tiket Masuk Event Dengan Memperhitungkan Tingkat Koreksi Kesalahan," *JATI (Jurnal Teknol. dan Informasi)*, vol. 7, no. 2, pp. 25–44, 2017.
- [5] V. Uzun, "QR-code based Hospital Systems for Healthcare in Turkey," in *Annual Computer Software and Applications Conference*, 2016, pp. 71–76.
- [6] F. Masalha and N. Hirzallah, "A Students Attendance System Using QR Code," *Int. J. Adv. Comput. Sci. Appl.*, vol. 5, no. 3, pp. 75–79, 2014.
- [7] P. Shrivastava and U. P. Singh, "Error Detection and Correction Using Reed Solomon Codes," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 8, pp. 965–969, 2013.
- [8] A. Farissi and M. Fachrurrozi, "Algoritma RSA Kombinasi dan Skema QR Code untuk Mengamankan Data Penjualan Tiket Online," in *Prosiding Annual Research Seminar 2017 Computer Science and ICT*, 2017, vol. 3, no. 1, pp. 3–7.
- [9] S. Tiwari, "An Introduction To QR Code Technology," *Int. Conf. Inf. Technol.*, vol. 1, pp. 39–44, 2016.
- [10] P. Sutheebanjard, "QR-Code Generator," *IEEE*, pp. 89–92, 2010.
- [11] N. Ani and R. Deby, "Implementasi QR Code pada Resep Dokter," in *Seminar Nasional Pengkajian dan Penerapan Teknologi Industri*, 2010, pp. 27–32.