

# Penerapan Metode Kompresi Huffman untuk Meningkatkan Jumlah Data Informasi pada Kode Quick Response

Hidayat\*<sup>1</sup>, Muchammad Yaasiin Asshobry<sup>2</sup>

<sup>1</sup>Teknik Komputer, <sup>2</sup>Sistem Komputer, Universitas Komputer Indonesia  
e-mail: <sup>1</sup>hidayat@email.unikom.ac.id, <sup>2</sup>yasinasshobry@gmail.com

\*Penulis Korespondensi

Diterima: 21 Agustus 2022; Direvisi: 03 November 2022; Disetujui: 07 November 2022

## Abstrak

Kecepatan memperoleh informasi merupakan hal yang sangat penting. Pemanfaatan kode Quick Response (QR) menjadi salah satu cara dalam upaya memperoleh informasi secara cepat. Namun, setiap versi kode QR Code sendiri memiliki batasan data informasi yang dapat ditampung, sehingga umumnya kode QR hanya menampung tautan website agar informasinya diperoleh melalui website. Peningkatan daya tampung data pada kode QR tentunya dapat memudahkan pengguna untuk menyimpan data informasi lebih banyak sehingga informasi dapat diperoleh tanpa mengakses website. Penelitian ini menerapkan metode kompresi Huffman untuk meningkatkan jumlah informasi yang dapat ditampung dalam kode QR. Kompresi Huffman merupakan algoritma kompresi lossless yang cocok digunakan pada data teks. Tahapan yang dilakukan pada penelitian ini terdiri atas analisis kebutuhan, perancangan, implementasi dan pengujian. Sistem yang dibangun terdiri atas dua aplikasi, yaitu aplikasi generator dan aplikasi pemindai kode QR. Aplikasi generator berfungsi menjalankan kompresi dan menghasilkan kode QR. Aplikasi pemindai kode QR berfungsi memindai kode QR dan melakukan dekompresi serta menampilkan informasi ke layar smartphone. Hasil penelitian menunjukkan kedua aplikasi dapat berfungsi dengan baik. Efisiensi hasil kompresi pada 30 data teks yang diuji mencapai rata-rata 56,6%. Hal ini menjadikan kode QR dapat menampung data informasi rata-rata dua kali lipat data teks tanpa kompresi.

**Kata kunci:** Data informasi, Kode QR, Kompresi data, Metode kompresi Huffman

## Abstract

The speed of obtaining information is very important. The use of Quick Response (QR) codes is one way to get information quickly. However, each QR Code version has a limit on the information data that can be accommodated, so generally, QR codes only accommodate website links so that information can be obtained through the website. Increasing the data capacity of the QR code can certainly make it easier for users to store more information data so that information can be obtained without accessing the website. This study applies the Huffman compression method to increase the amount of information that can be accommodated in a QR code. Huffman compression is a lossless compression algorithm that is suitable for use on text data. The stages carried out in this research consist of needs analysis, design, implementation and testing. The system built consists of two applications, namely a generator application and a QR code scanner application. The generator application functions to run compression and generate QR codes. The QR code scanner application functions to scan QR codes and decompresses and displays information to the smartphone screen. The results showed that both applications can function properly. The efficiency of the compression results on 30 tested text data reached an average of

---

56.6%. It makes the QR code able to accommodate an average of twice as much information data without compression.

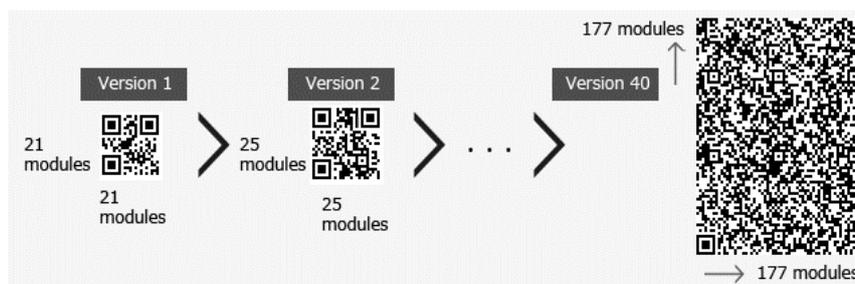
**Keywords:** Data compression, Data information, QR code, Huffman compression method

## 1. PENDAHULUAN

Kecepatan mendapatkan informasi yang lengkap sangatlah penting. Sebagai contoh papan iklan yang terpampang di samping jalan merupakan sebuah sarana informasi dapat dilihat oleh setiap orang yang melewati jalan tersebut. Umumnya, papan reklame memuat banyak gambar atau ilustrasi dan sedikit kata-kata atau kalimat singkat, sehingga informasi pada papan iklan tersebut cenderung bersifat umum. Untuk mengetahui informasi lebih detail, pembaca biasanya mencari informasi lebih lanjut dengan mengakses laman internet tertentu sehingga diperlukan waktu yang lebih banyak untuk menelusuri informasi tersebut. Padahal adakalanya pembaca memerlukan informasi yang cepat diterima dan praktis. Oleh karena itu, diperlukan suatu cara agar dapat menampilkan informasi yang lebih detail yang bersumber dari papan iklan tanpa menjelajah halaman webnya.

Salah satu cara yang umum digunakan untuk dapat mendapat informasi yang lebih banyak dari sebuah gambar adalah penggunaan teknologi *Quick Response code* atau dikenal QR code. Kode QR merupakan teknologi yang sangat cepat diterima keberadaannya oleh pengguna. Pertumbuhan cepat ini seiring dengan semakin banyaknya pengguna smartphone yang memanfaatkan teknologi kode QR dalam berbagai bidang [1]. Sebagai contoh, pada penelitian [2]–[5] dipaparkan pemanfaatan kode QR dalam melakukan pembayaran dan informasi struk belanja. Selain itu, penelitian [6]–[8] menjelaskan pemanfaatan kode QR dalam bidang pendidikan dan museum. Sementara itu, penelitian [9]–[11] memaparkan pemanfaatan kode QR untuk memberikan informasi pada bidang kesehatan. Selanjutnya, penelitian [12] pemanfaatan kode QR juga digunakan pada produk agar dapat dilacak perpindahannya dan juga pada makanan [13]. Penggunaan kode QR memiliki manfaat yang besar dalam berbagai bidang ilmu.

Kode QR sendiri dikembangkan pada tahun 1994 oleh Denso Wave Incorporation yang merupakan sebuah perusahaan otomotif asal Jepang. Kode QR sendiri memiliki ukuran atau versi yang berbeda-beda dengan daya tampung informasi yang berbeda-beda pula. Terdapat 40 versi kode QR yang masing-masing menunjukkan jumlah data yang dapat disimpan. Setiap versi kode QR memiliki kapasitas data maksimum yang berbeda dan juga tingkat koreksi kesalahannya [14]. Semakin rendah versi kode QR maka semakin sedikit jumlah data yang dapat disimpan. Sebaliknya, semakin tinggi versi kode QR semakin besar data yang dapat ditampung. Hal ini menyebabkan kode QR yang dihasilkan berukuran lebih besar seperti ditunjukkan pada Gambar 1.



Gambar 1. Contoh versi kode QR [14]

Umumnya, pemanfaatan kode QR versi pada papan reklame digunakan untuk menyimpan tautan website agar pengguna dapat mengakses website tersebut untuk memperoleh informasi secara lengkap [15]. Hal ini tentunya harus didukung akses internet yang stabil agar

tautan website tersebut dapat diakses. Pastinya, proses mengakses tersebut akan membutuhkan waktu sehingga informasi yang diperoleh belum tentu secepat yang diharapkan pengguna. Hal ini mendorong peneliti untuk meningkatkan jumlah informasi pada kode QR agar informasi yang dapat diterima oleh pengguna akan lebih banyak dari informasi kode QR pada umumnya tanpa mengakses website reklame tersebut.

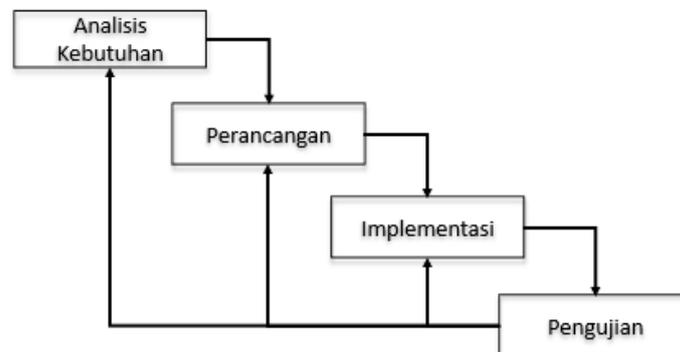
Penelitian ini dilakukan untuk meningkatkan kapasitas informasi pada kode QR dengan memanfaatkan metode kompresi. Kompresi berarti pengkodean informasi menggunakan bit lebih sedikit daripada representasi asli untuk mengurangi konsumsi sumber daya, seperti penyimpanan dan bandwidth atau bahkan untuk alasan keamanan [16]–[18]. Kompresi dapat berupa lossy atau lossless. Kompresi lossy mengurangi bit dengan menghapus informasi yang tidak perlu atau kurang penting, sedangkan kompresi lossless mengidentifikasi dan menghilangkan redundansi statistik. Pada metode kompresi lossless tidak ada informasi yang hilang. Data asli direkonstruksi seluruhnya dari data terkompresi, sedangkan kompresi lossy memungkinkan rekonstruksi hanya perkiraan data asli. Pada penelitian ini akan menggunakan metode kompresi Huffman yang merupakan kompresi lossless dan banyak dimanfaatkan pada teks seperti pada penelitian [19].

Metode kompresi Huffman termasuk masih populer karena implementasinya yang sederhana dan kompresi yang lebih cepat. Pada kompresi Huffman, simbol atau karakter yang memiliki probabilitas paling besar akan diberikan jumlah bit kode yang sedikit, sedangkan simbol atau karakter dengan probabilitas paling kecil akan memperoleh jumlah bit kode yang banyak. Kode tersebut diperoleh dengan cara menyusun sebuah pohon Huffman untuk masing-masing simbol atau karakter berdasarkan nilai probabilitasnya.

## 2. METODE PENELITIAN

### 2.1. Tahapan Penelitian

Tahapan penelitian pada aplikasi ini menggunakan metode waterfall. Tahapan dimulai dari Analisis kebutuhan, Perancangan, Implementasi, dan Validasi atau Pengujian. Pada tahapan analisis dilakukan pengumpulan informasi yang dibutuhkan oleh sistem yang akan dibangun. Selanjutnya, informasi tersebut didefinisikan sebagai spesifikasi sistem.



Gambar 2. Tahapan penelitian pada pengembangan aplikasi [20]

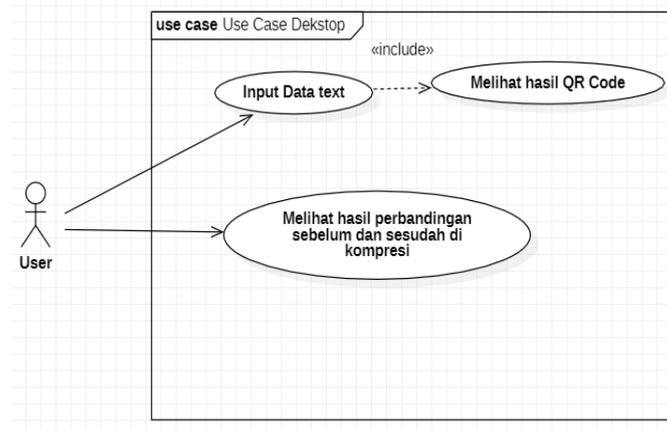
Pada tahapan perancangan dilakukan perancangan sistem berdasarkan hasil analisis kebutuhan. Pada tahapan ini memuat tahapan perancangan antarmuka aplikasi dan tahapan pemrograman. Tahapan selanjutnya adalah implementasi. Pada tahapan ini dilakukan implementasi hasil rancangan menjadi sistem yang dapat diaplikasikan. Tahapan terakhir adalah tahapan pengujian, yaitu tahapan untuk melakukan pengujian sistem untuk mengetahui berfungsi atau tidaknya setiap bagian dalam sistem. Tahapan penelitian ini diperlihatkan pada Gambar 2.

### 2.2. Perancangan

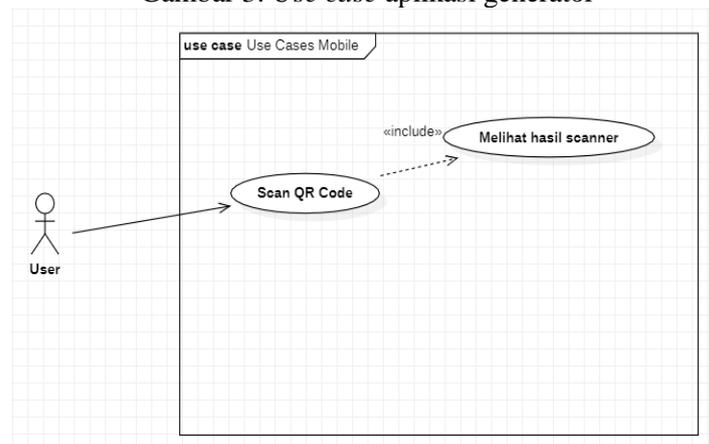
Perancangan sistem terdiri pada dua bagian, yaitu perancangan pada aplikasi generator kompresi dan kode QR dan aplikasi pemindai kode QR. Aplikasi generator dirancang dengan menggunakan platform desktop. Aplikasi generator kode QR dibangun dengan menggunakan bahasa pemrograman Java melalui program aplikasi NetBeans IDE 8.2. Pada aplikasi tersebut terdapat fungsi untuk melakukan kompresi data dengan metode Huffman. Aplikasi akan secara otomatis melakukan kompresi ketika pengguna memasukkan data informasi dan menekan tombol *Generate*. Kompresi data dilakukan pada data teks berbahasa Indonesia. Kamus data hasil pembentukan metode pengkodean Huffman disimpan pada program secara tetap. Selain itu, pada aplikasi ini juga terdapat fungsi untuk menghasilkan kode QR dari data hasil kompresi dengan menggunakan *Nayuki library*. Pada aplikasi kedua, yaitu aplikasi pemindai kode QR, dibangun dengan menggunakan bahasa pemrograman Java pada platform android melalui program aplikasi Android Studio. Pada aplikasi ini terdapat fungsi untuk melakukan dekompresi data secara otomatis ketika pengguna mengarahkan kamera smartphone ke arah kode QR.

### 2.2.1. Use Case Diagram

*Use case diagram* yang memodelkan aplikasi yang akan dibangun diperlihatkan pada Gambar 3 dan Gambar 4. Gambar 3 memperlihatkan interaksi pengguna yang akan mengubah data informasi menjadi kode QR. Langkah pertama yang dilakukan oleh pengguna adalah membuka aplikasi, kemudian pengguna memilih dokumen yang akan diubah menjadi kode QR, dan diakhiri dengan menekan tombol *Generate*. Selanjutnya, pengguna dapat melihat hasilnya pada folder kode QR. Gambar 4 memperlihatkan pengguna memindai kode QR menjadi data informasi yang ingin diketahui. Untuk melakukan pemindaian, pengguna hanya perlu memfokuskan kamera *smartphone* pada kode QR, kemudian menunggu beberapa saat hingga hasil pemindaian tampil pada layar.



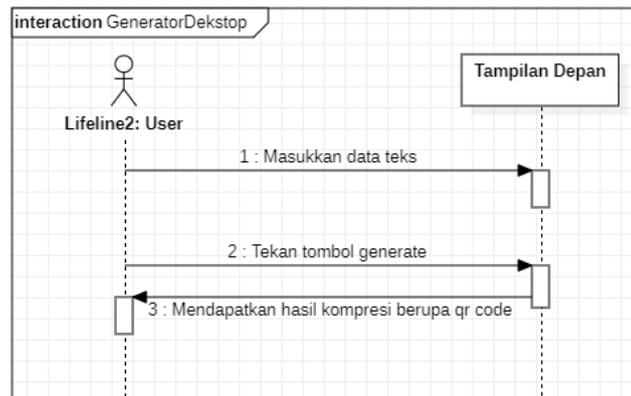
Gambar 3. Use case aplikasi generator



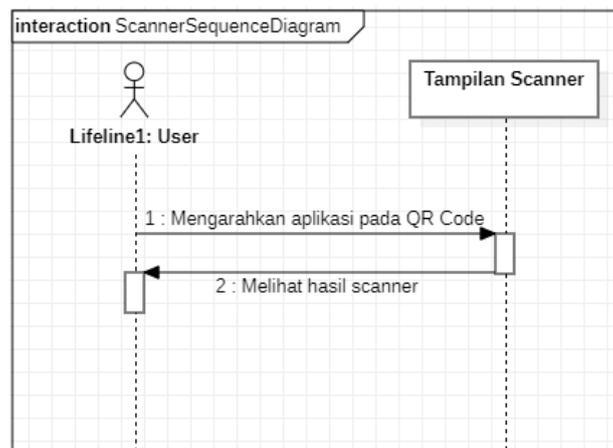
Gambar 4. Use case aplikasi pemindai

2.2.2. Sequence Diagram

Sequence Diagram aplikasi Generator kode QR diperlihatkan pada Gambar 5. Gambar tersebut memperlihatkan waktu pemakaian pengguna terhadap aplikasi generator. Gambar sequence tersebut menunjukkan bahwa penggunaan aplikasi sangat mudah. Proses pengubahan data informasi menjadi kode QR diperoleh setelah pengguna memilih data dengan format .txt, kemudian menekan tombol generate. Selanjutnya, pengguna dapat mengambil hasilnya pada folder yang telah ditentukan. Sementara itu, Sequence Diagram untuk aplikasi pemindai kode QR diperlihatkan pada Gambar 6. Gambar tersebut memperlihatkan waktu penggunaan aplikasi pemindai yang cukup singkat. Pengguna hanya perlu memfokuskan kamera *smartphone* pada kode QR untuk melihat hasil pemindaian.



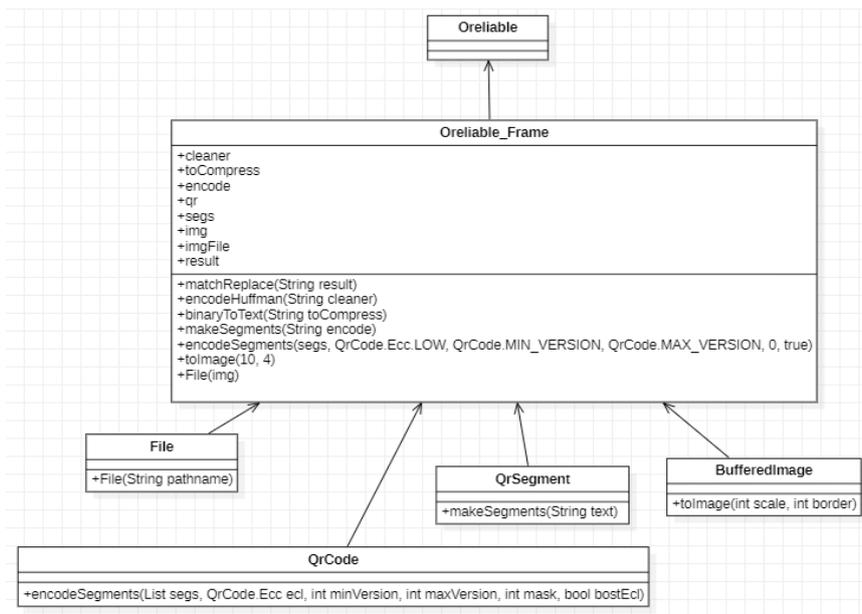
Gambar 5. Sequence diagram aplikasi generator kode QR



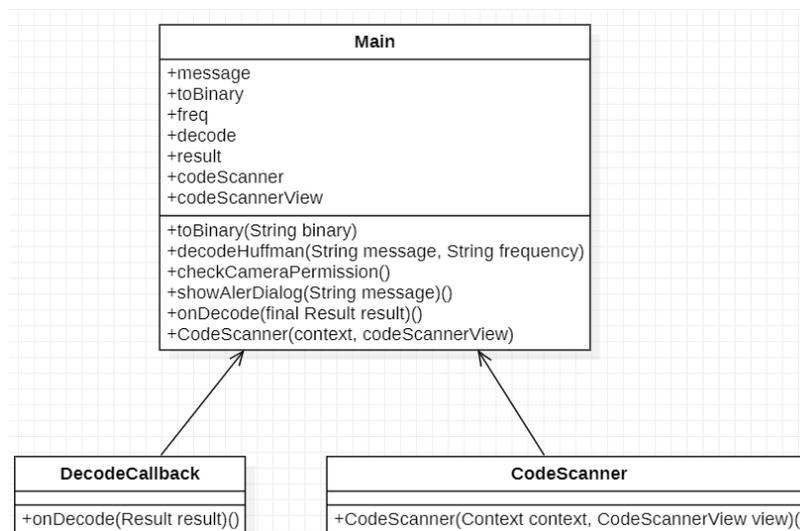
Gambar 6. Sequence diagram aplikasi pemindai kode QR

2.2.3. Class Diagram

Class diagram pada aplikasi generator ditunjukkan pada Gambar 7. Pada Class diagram pada aplikasi generator terdapat satu kelas utama dan empat kelas penunjang, yaitu *File*, *QrSegment*, *BufferedImage* dan *QrCode*. Kelas *File* memiliki fungsi untuk mengambil dan menulis informasi pada file berformat txt. Kelas *QrSegment* berfungsi untuk menentukan jenis informasi yang akan diubah menjadi kode QR. Kelas *QrCode* berfungsi untuk mengubah data informasi menjadi kode QR dengan memanfaatkan *BufferedImage* sebagai konversi menjadi data gambar yang bisa dilihat dan digunakan pengguna.



Gambar 7. Class diagram aplikasi generator



Gambar 8. Class diagram aplikasi pemindai

*Class diagram* pada aplikasi pemindai kode QR ditunjukkan pada Gambar 8. *Class diagram* pada aplikasi pemindai kode QR memiliki kelas-kelas yang menjadi dasar pembangunan aplikasi pemindai. Aplikasi pemindai terdiri dari satu kelas utama dan dua kelas penunjang. Kelas pertama yang akan dipanggil adalah *CodeScanner* yang berfungsi untuk memanggil fungsi kamera lalu mengambil dan mengubah nilai dari kode QR menjadi data informasi yang sebelumnya telah dilakukan kompresi. Selanjutnya, kelas *DecodeCallback* akan melakukan dekomposisi data hasil pemindaian menjadi data informasi yang utuh.

### 3. HASIL DAN PEMBAHASAN

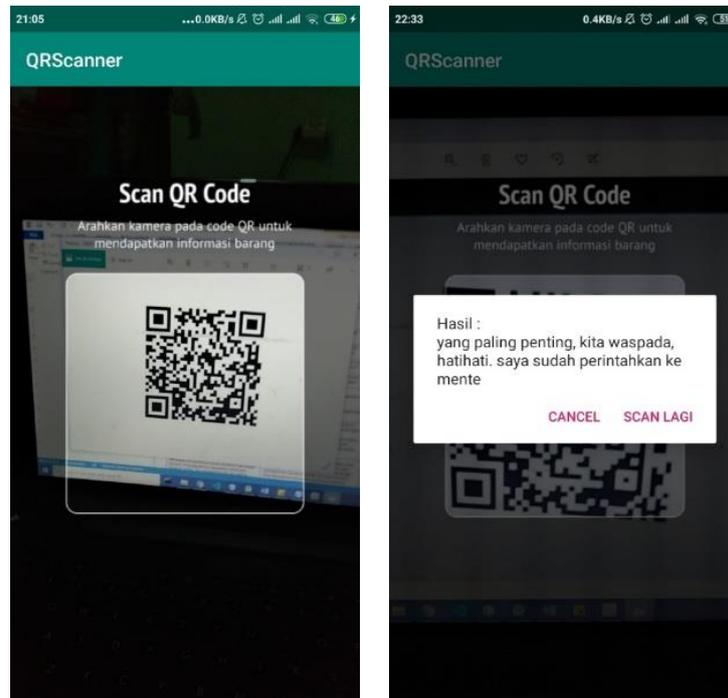
Pada penelitian ini, aplikasi generator diterapkan pada laptop Asus X454Y dan aplikasi pemindai kode QR menggunakan *smartphone* Redmi 7A. Tampilan aplikasi Generator untuk melakukan kompresi data dan menghasilkan kode QR diperlihatkan pada Gambar 9. Proses

kompresi data menggunakan metode Huffman dengan memanfaatkan *library* yang diperoleh secara *open source*.

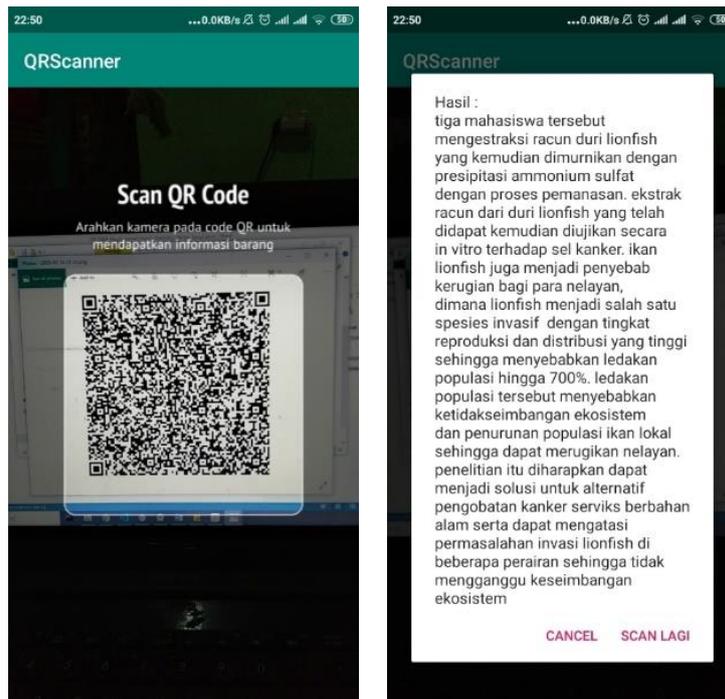


Gambar 9. Tampilan Aplikasi generator kompresi dan kode QR

Sementara itu, contoh tampilan aplikasi pemindai kode QR untuk menampilkan informasi dari kode QR ditampilkan pada Gambar 10 dan Gambar 11. Pada aplikasi ini, pengguna hanya perlu memfokuskan kamera smartphone pada kode QR dan akan memperoleh hasilnya pada kotak dialog di layar *smartphone*. Jika selesai, pengguna dapat menekan tombol *Cancel* dan jika ingin pemindaian kembali, pengguna dapat menekan tombol *Scan Lagi*.



Gambar 10. Contoh tampilan pada aplikasi pemindai data text7.txt



Gambar 11. Contoh tampilan pada aplikasi pemindai data text17.txt

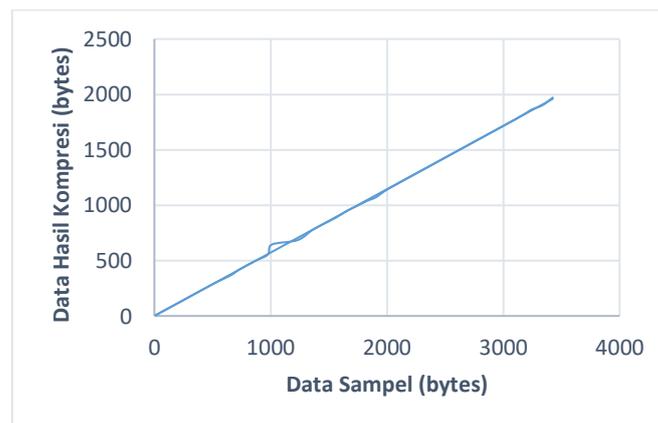
### 3.1. Pengujian

Pengujian dilakukan pada 30 dokumen dengan format .txt yang memiliki ukuran yang berbeda-beda. Hasil pengujian ditunjukkan pada Tabel 1. Setiap dokumen teks berhasil dilakukan kompresi dan dekompresi dengan baik. Rasio kompresi pada data Tabel 1 merupakan hasil pembagian ukuran keluaran hasil kompresi dibagi ukuran asli sebelum dilakukan kompresi seperti pada persamaan (1). Nilai ini menunjukkan nilai efisiensi pemanfaatan metode kompresi Huffman pada sistem.

$$\text{Compression Ratio} = \frac{\text{size of the output stream}}{\text{size of the input stream}} \quad (1)$$

Nilai efisiensi terkecil terjadi pada text9.txt yaitu sebesar 47%. Adapun nilai efisiensi tertinggi terjadi pada text19.txt yaitu sebesar 63%. Rata-rata nilai efisiensi hasil pengujian secara keseluruhan adalah 56,6%.

Hasil pengujian dalam bentuk grafik dapat dilihat pada Gambar 12. Nilai efisiensi hasil kompresi berubah secara linier.

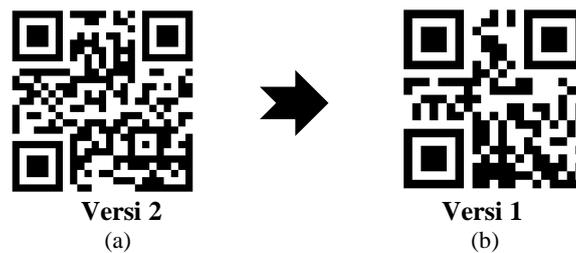


Gambar 12. Hubungan data sampel dengan data hasil pengujian

### 3.2. Pengujian QR Code

Pada pengujian perubahan kode QR dapat dilihat contoh perbedaan kode QR data teks (*text2.txt*) yang berukuran 20 bytes sebelum dan sesudah proses kompresi pada Gambar 13. Gambar tersebut menunjukkan bahwa data teks sebelum proses kompresi memerlukan kode QR dengan versi 2 (Gambar 13a). Sementara itu, kode QR yang dihasilkan setelah data teks melalui proses kompresi adalah kode QR versi 1 (Gambar 13b) karena ukuran data teks yang dihasilkan sebesar 11 bytes. Hal ini menunjukkan bahwa data informasi hasil kompresi dapat ditampung pada kode QR yang lebih rendah atau versi lebih kecil.

Hasil pengujian secara keseluruhan menunjukkan bahwa aplikasi yang dirancang dapat meningkatkan jumlah informasi yang disimpan pada kode QR mencapai rata-rata sebesar 56,6%. Data informasi yang digunakan pada penelitian ini masih terbatas pada teks bahasa Indonesia sehingga pada penggunaan data teks berbahasa lain akan memiliki hasil yang berbeda. Hal ini disebabkan kamus kompresi yang digunakan bersifat statis.



Gambar 13. Contoh kode QR, a) kode QR sebelum proses kompresi, b) kode QR setelah proses kompresi

Tabel 1. Hasil pengujian kompresi data teks dan rasio kompresi

No	Data Sampel	Ukuran asli (bytes)	Ukuran hasil kompresi (bytes)	Rasio Kompresi
1	<i>text1.txt</i>	10	5	50%
2	<i>text2.txt</i>	20	11	55%
3	<i>text3.txt</i>	30	17	56%
4	<i>text4.txt</i>	43	25	62%
5	<i>text5.txt</i>	50	30	60%
6	<i>text6.txt</i>	61	36	59%
7	<i>text7.txt</i>	81	46	57%
8	<i>text8.txt</i>	85	50	58%
<b>9</b>	<b><i>text9.txt</i></b>	<b>95</b>	<b>54</b>	<b>47%</b>
10	<i>text10.txt</i>	102	59	57%
11	<i>text11.txt</i>	237	135	56%
12	<i>text12.txt</i>	321	184	57%
13	<i>text13.txt</i>	447	256	57%
14	<i>text14.txt</i>	543	309	57%
15	<i>text15.txt</i>	660	369	55%
16	<i>text16.txt</i>	735	422	57%
17	<i>text17.txt</i>	871	499	57%
18	<i>text18.txt</i>	976	554	56%
<b>19</b>	<b><i>text19.txt</i></b>	<b>1010</b>	<b>645</b>	<b>63%</b>
20	<i>text20.txt</i>	1233	685	55%
21	<i>text21.txt</i>	1365	780	57%

No	Data Sampel	Ukuran asli (bytes)	Ukuran hasil kompresi (bytes)	Rasio Kompresi
22	text22.txt	1560	887	56%
23	text23.txt	1655	949	57%
24	text24.txt	1731	987	57%
25	text25.txt	1825	1037	56%
26	text26.txt	1908	1073	56%
27	text27.txt	2021	1153	57%
28	text28.txt	3106	1776	57%
29	text29.txt	3232	1857	57%
30	text30.txt	3338	1903	57%
Rata-rata				56,6%

#### 4. KESIMPULAN

Pemanfaatan metode kompresi Huffman pada data informasi yang disimpan pada kode QR telah menunjukkan peningkatan data informasi. Sistem yang dibangun menghasilkan aplikasi generator kode QR dan aplikasi pemindai kode QR yang dapat melakukan kompresi dan dekompresi data informasi untuk data teks dalam Bahasa Indonesia dengan kamus kompresi data yang digunakan pada perancangan ini bersifat statis. Hasil pengujian menunjukkan bahwa aplikasi yang dibangun dapat meningkatkan jumlah informasi yang ditampung pada kode QR dengan rata-rata hasil kompresinya sebesar 57%. Peningkatan terbesar terdapat pada text19.txt dengan prosentase peningkatan sebesar 63%. Adapun peningkatan terkecil adalah sebesar 47% yang terdapat pada text9.txt. Hasil ini menunjukkan bahwa aplikasi yang dirancang telah mampu meningkatkan jumlah informasi yang dapat disimpan pada kode QR. Penelitian yang dilakukan masih terbatas menggunakan bahasa Indonesia sehingga jika diterapkan pada teks dengan bahasa lain maka hasilnya tentu akan berbeda karena kamus kompresi yang digunakan adalah kamus kompresi yang bersifat statis untuk data teks berbahasa Indonesia. Pada aplikasi yang telah dirancang ini masih dapat dilakukan pengembangan untuk menerapkan kamus kompresi data yang dinamis sehingga aplikasi dapat diimplementasikan pada berbagai bahasa yang berbeda.

#### DAFTAR PUSTAKA

- [1] S. Tiwari, "An introduction to QR code technology," in *Proceedings - 2016 15th International Conference on Information Technology, ICIT 2016*, 2017, vol. 1, pp. 39–44, doi: 10.1109/ICIT.2016.38.
- [2] L. Yan, G. W.-H. Tan, X.-M. Loh, J.-J. Hew, and K.-B. Ooi, "QR code and mobile payment: The disruptive forces in retail," *J. Retail. Consum. Serv.*, vol. 58, 2021, doi: 10.1016/j.jretconser.2020.102300.
- [3] N. P. A. Karniawati, G. S. Darma, L. P. Mahyuni, and I. G. Sanica, "Community perception of using qr code payment in era new normal," *Palarch's J. Archaeol. Egypt/Egyptology*, vol. 18, no. 1, pp. 3986–3999, 2021.
- [4] H. Alamoudi, "Examining Retailing Sustainability in the QR Code-Enabled Mobile Payments Context During the COVID-19 Pandemic," *Int. J. Cust. Relatsh. Mark. Manag.*, vol. 13, no. 1, pp. 1–22, 2021, doi: 10.4018/ijcrmm.289210.
- [5] A. R. Wardana and H. Hidayat, "Rancang Bangun Aplikasi Kasir Pencetak Struk Belanja Kode Quick Response berbasis Website dan Android," *JOINS (Journal Inf. Syst.)*, vol. 5, no. 2, pp. 208–220, 2020, doi: 10.33633/joins.v5i2.3809.
- [6] M. Smith, M. Segura-Totten, and K. West, "QR Code Lecture Activity as a Tool for

- Increasing Nonmajors Biology Students' Enjoyment of Interaction with Their Local Environment," *J. Microbiol. Biol. Educ.*, vol. 19, no. 1, pp. 1–6, 2018, doi: 10.1128/jmbe.v19i1.1453.
- [7] S. R. Mogali, R. Vallabhajosyula, C. H. Ng, D. Lim, E. T. Ang, and P. Abrahams, "Scan and Learn: Quick Response Code Enabled Museum for Mobile Learning of Anatomy and Pathology," *Anat. Sci. Educ.*, vol. 12, no. 6, pp. 664–672, 2019, doi: 10.1002/ase.1848.
- [8] S. Goyal, S. Yadav, and M. Mathuria, "Exploring concept of QR code and its benefits in digital education system," in *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*, 2016, pp. 1141–1147, doi: 10.1109/ICACCI.2016.7732198.
- [9] C. T. Karia, A. Hughes, and S. Carr, "Uses of quick response codes in healthcare education: A scoping review," *BMC Med. Educ.*, vol. 19, no. 1, pp. 1–14, 2019, doi: 10.1186/s12909-019-1876-4.
- [10] J. T. Jamu, H. Lowi-Jones, and C. Mitchell, "Just in time? Using QR codes for multi-professional learning in clinical practice," *Nurse Educ. Pract.*, vol. 19, pp. 107–112, 2016, doi: 10.1016/j.nepr.2016.03.007.
- [11] Y. Y. Chang *et al.*, "A mobile medical QR-code authentication system and its automatic FICE image evaluation application," *J. Appl. Res. Technol.*, vol. 13, no. 2, pp. 220–229, 2015, doi: 10.1016/j.jart.2015.06.020.
- [12] L. Tarjan, I. Šenk, S. Tegeltija, S. Stankovski, and G. Ostojic, "A readability analysis for QR code application in a traceability system," *Comput. Electron. Agric.*, vol. 109, pp. 1–11, 2014, doi: 10.1016/j.compag.2014.08.015.
- [13] K. Rotsios, A. Konstantoglou, D. Folinias, T. Fotiadis, L. Hatzithomas, and C. Boutsouki, "Evaluating the Use of QR Codes on Food Products," *Sustainability*, vol. 14, no. 8, pp. 1–15, 2022, doi: 10.3390/su14084437.
- [14] DENSO WAVE, "Information capacity and versions of QR Code," *QRcode.com*, 2014. <http://www.qrcode.com/en/about/version.html> (accessed Jan. 10, 2019).
- [15] J. Cpajak and A. Gavrovskaa, "Smart Web-based Advertising System using Eye-tracking Technology," in *2021 29th Telecommunications Forum (TELFOR)*, 2021, pp. 1–4, doi: <https://doi.org/10.1109/TELFOR52709.2021.9653350>.
- [16] U. Jayasankar, V. Thirumal, and D. Ponnurangam, "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 33, no. 2, pp. 119–140, 2021, doi: 10.1016/j.jksuci.2018.05.006.
- [17] S. Hamdan, A. Awaian, and S. Almajali, "Compression techniques used in IoT: A comparative study," in *2019 2nd International Conference on New Trends in Computing Sciences, ICTCS 2019 - Proceedings*, 2019, pp. 1–5, doi: 10.1109/ICTCS.2019.8923112.
- [18] J. S. Resende, P. R. Sousa, R. Martins, and L. Antunes, "Breaking MPC implementations through compression," *Int. J. Inf. Secur.*, vol. 18, no. 4, pp. 505–518, 2019, doi: 10.1007/s10207-018-0424-2.
- [19] J. M. Cadavid and H. D. V. Madrigal, "A lossless compression method for chat messages based on huffman coding and dynamic programming," *Computers*, vol. 10, no. 3, pp. 1–14, 2021, doi: 10.3390/computers10030028.
- [20] A. Mishra and D. Dubey, "A Comparative Study of Different Software Development Life Cycle Models in Different Scenarios," *Int. J. Adv. Res. Comput. Sci. Manag. Stud.*, vol. 1, no. 5, pp. 2321–7782, 2013, [Online]. Available: <http://www.ijarcsms.com/docs/paper/volume1/issue5/V1I5-0008.pdf>.