

# Pengembangan Aplikasi QR Code Generator dan QR Code Reader dari Data Berbentuk Image

M. Pasca Nugraha  
 Institut Teknologi Bandung  
 Jl. Ganesha 10 Bandung 40132  
 lf17033@students.if.itb.ac.id

Dr. Ir. Rinaldi Munir M.T.  
 Institut Teknologi Bandung  
 Jl. Ganesha 10 Bandung 40132  
 rinaldi@informatika.org

## ABSTRACT

*QR Code*, kependekan dari *Quick Response Code*, merupakan gambar dua dimensi yang memiliki kemampuan untuk menyimpan data. *QR Code* biasa digunakan untuk menyimpan data berupa teks, baik itu numerik, alfanumerik, maupun kode biner. *QR Code* banyak digunakan untuk keperluan komersil, khususnya di Jepang, biasanya berisi link url ke alamat tertentu atau sekedar teks berisi iklan, promosi, dan lain-lain. Salah satu hal yang belum umum digunakan pada *QR Code* adalah menyisipkan gambar pada informasi yang disimpannya. Hal ini dapat menambah daya tarik pembaca terutama untuk urusan iklan, poster, dan komersil lainnya.

Pada tugas akhir ini dilakukan studi mengenai kemungkinan membuat *QR Code* dari data berbentuk *image*. Oleh karena itu dilakukan analisis bagaimana menjadikan gambar sebagai isi dari informasi yang dikandung oleh *QR Code*. Solusi yang ditawarkan adalah dengan mengubah gambar menjadi representasi lain yang dapat dimengerti oleh *QR Code generator* dan *QR Code reader*, yaitu alfanumerik. Sebelumnya file gambar tersebut diubah terlebih dahulu menjadi *stream* byte.

Tugas akhir ini mengimplementasikan pengembangan aplikasi *QR Code Generator* dan *QR Code Reader*, dimana data masukannya berupa *image*, bukan teks seperti biasanya. Kemudian dilakukan pengujian apakah feasible untuk diterapkan di dunia nyata. Pengujian juga dilakukan untuk mengetahui berapa ukuran maksimum file gambar yang dapat dijadikan *QR Code* dan apakah terjadi perubahan kualitas gambar sebelum dijadikan *QR Code* dengan setelah dibaca dan diterjemahkan kembali dari *QR Code*. Implementasi perangkat lunak dilakukan dengan menggunakan bahasa C# dengan framework .NET. Sedangkan kakas yang digunakan adalah Microsoft Visual Studio 2008.

Berdasarkan pengujian yang dilakukan, aplikasi *QR Code* dari data berbentuk *image* ini tidak feasible untuk diterapkan di dunia nyata, karena ukuran gambar maksimum yang dapat dijadikan *QR Code* terlalu kecil. Secara visual memang tidak ada perubahan kualitas gambar sebelum dan setelah proses *encoding-decoding QR Code* walaupun dari segi ukuran file ada sedikit perubahan.

**Keywords**—*QR Code*, *image*, *QR Code generator*, *QR Code reader*.

## 1. PENDAHULUAN

Teknologi informasi memberikan pengaruh yang sangat besar pada kehidupan manusia hampir di setiap bidang kehidupan [4]. Contohnya adalah pada bidang perindustrian, yang dengan diterapkannya teknologi informasi terbukti meningkatkan

efektivitas dan efisiensi perusahaan [14]. Salah satu teknologi yang cukup dikenal adalah *barcode*.

Pada masa ini, penggunaan *barcode* sudah tidak asing lagi di industri di seluruh dunia. Hal ini adalah untuk memudahkan pelaku industri dalam mengelola inventori yang mereka miliki, karena *barcode* ini menyimpan data spesifik seperti kode produksi, nomor identitas, dan lain-lain sehingga sistem komputer dapat mengidentifikasi informasi yang dikodekan dalam *barcode* dengan mudah [16]. Seiring dengan perkembangan teknologi yang begitu pesatnya, penggunaan *barcode* kini mulai digantikan dengan *QR Code*.

*QR Code* adalah *image* dua dimensi yang merepresentasikan suatu data, terutama data berbentuk teks. *QR Code* merupakan evolusi dari *barcode* yang awalnya satu dimensi menjadi dua dimensi. *QR Code* memiliki kemampuan menyimpan data yang lebih jauh besar daripada *barcode*.

Saat ini penggunaan *QR Code* sudah cukup luas. Banyak negara di dunia, terutama Jepang, telah menerapkan teknologi *QR Code* pada perindustriannya. Sementara di Indonesia, *QR Code* sudah diterapkan pada beberapa perusahaan. Salah satunya adalah pada surat kabar Kompas, yang mengklaim sebagai pelopor penggunaan *QR Code* di Indonesia, yang diterbitkan oleh Kelompok Kompas Gramedia [10]. Hal ini terlihat pada surat kabar tersebut dimana terdapat *QR Code* yang merepresentasikan artikel pada surat kabar tersebut.

Namun hingga pada saat ini, *QR Code* baru digunakan terbatas untuk menyimpan data teks seperti alamat URL, nomor telepon, informasi, dan lain-lain. Belum ada aplikasi maupun penelitian untuk menyimpan bentuk data lain pada *QR Code*. Padahal *QR Code* banyak digunakan untuk keperluan komersil seperti iklan, sedangkan iklan yang ditambahkan data lain selain teks, misalnya gambar, akan menambah daya tarik konsumen.

Oleh karena itu, dari latar belakang di atas, perlu dilakukan studi dan analisis mengenai kemungkinan pembangkitan *QR Code* dari data berbentuk gambar. Studi tersebut diimplementasikan menjadi sebuah perangkat lunak *QR Code generator* dan *QR Code reader* untuk kemudian dilakukan pengujian terhadap perangkat lunak tersebut. Dari pengujian yang dilakukan dapat diketahui kelayakan data gambar disimpan dalam *QR Code*.

## 2. STUDI LITERATUR

### 2.1 QR Code

*QR Code* adalah *image* berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. *QR Code* merupakan evolusi dari kode batang (*barcode*). *Barcode*

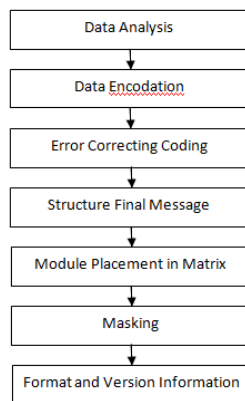
merupakan sebuah simbol penandaan objek nyata yang terbuat dari pola batang-batang berwarna hitam dan putih agar mudah untuk dikenali oleh komputer [12]. Contoh sebuah *QR Code* dapat dilihat pada gambar 1.



Gambar 1. Contoh *QR Code*

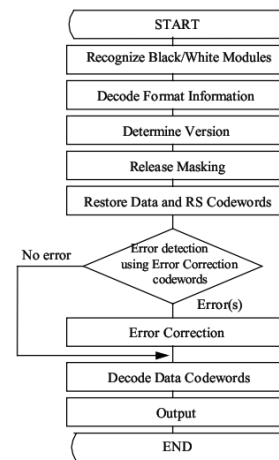
*QR Code* merupakan singkatan dari *Quick Response Code*, atau dapat diterjemahkan menjadi kode respon cepat. *QR Code* dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. *QR Code* ini dipublikasikan pada tahun 1994 dengan tujuan untuk pelacakan kendaraan di bagian manufaktur dengan cepat dan mendapatkan respon dengan cepat pula.

Prosedur pembangkitan *QR Code* dari sebuah teks dapat dijelaskan dengan diagram alir pada gambar 2.



Gambar 2. Diagram alir proses pembangkitan *QR Code*

Langkah-langkah untuk membaca *QR Code* menjadi teks aslinya merupakan *reverse* atau kebalikan dari langkah-langkah pada pembangkitan *QR Code*. Secara umum prosedur pembacaan *QR Code* dapat dijelaskan dengan diagram alir pada gambar 3.



Gambar 3. Diagram alir proses pembacaan *QR Code* [5]

Seiring berkembangnya *QR Code*, semakin banyak penelitian yang dilakukan mengenai kode simbol ini. Berbagai penelitian terus dilakukan, baik untuk menambah jumlah data yang dapat disimpan dalam *QR Code*, menambah resistensi terhadap kerusakan, dan lain-lain. Beberapa penelitian yang telah dilakukan diantaranya adalah :

1. Pembuatan aplikasi pembacaan *QR Code* menggunakan perangkat *mobile* berbasis J2ME [12]

Pada penelitian ini dibuat aplikasi pembaca *QR Code* menggunakan perangkat *mobile* berbasis sistem operasi J2ME dengan kamera digital yang terintegrasi pada *mobile* tersebut. Pada penelitian ini juga dianalisis tingkat akurasi pembaca *QR Code* tersebut sehingga ditemukan parameter yang mempengaruhi tingkat akurasinya.

2. *QR Code* untuk tandatangan digital [1]

Pada penelitian ini, peneliti menggunakan *QR Code* untuk tanda tangan digital. Data yang di-*encode* adalah *message digest* dari artikel atau tulisan yang akan dibuat tanda tangan digitalnya. Pada penelitian ini juga dibuat sebuah perangkat lunak yang langsung men-*generate QR Code* dari masukan berupa tulisan/artikel yang akan dibuat tanda tangan digitalnya.

3. *QR Code* untuk autentikasi novel *user* [7]

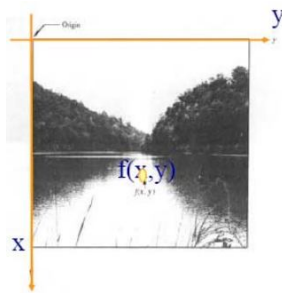
Pada penelitian ini, *QR Code* digunakan sebagai autentikasi *user* pada sebuah jaringan internet untuk *mobile phone*.

4. *QR Code* untuk edukasi [8]

Pada penelitian ini dijelaskan manfaat *QR Code* untuk edukasi, karena peneliti mengungkapkan bahwa selama ini *QR Code* kebanyakan hanya digunakan untuk kepentingan komersil. Contohnya adalah penggunaan *QR Code* untuk katalog perpustakaan.

## 2.2 Citra Digital

Citra dapat didefinisikan sebagai fungsi dua dimensi  $f(x,y)$  di mana  $x$  dan  $y$  adalah elemen citra yang disebut dengan piksel dan amplitudo  $f$  pada setiap pasang  $(x,y)$  disebut intensitas (*gray level*). Jika  $x$  dan  $y$  berhingga (*finite*) dan diskrit (tidak kontinu) maka disebut citra digital [13].

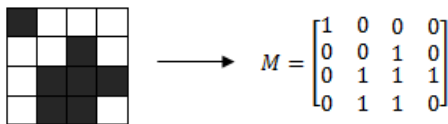


**Gambar 4.2** Contoh citra digital dalam  $f(x,y)$  [11]

Citra digital terdiri atas piksel-piksel berukuran kecil yang membentuk sebuah bentuk gambar yang dapat dilihat dengan mata. Semakin banyak piksel dalam sebuah gambar citra digital, maka kualitas gambar tersebut akan semakin baik. Kepadatan piksel-piksel yang ada dalam gambar ini disebut resolusi. Contoh citra digital dapat dilihat pada gambar 4.

### 2.2.1 Representasi Citra Digital

Data pada sebuah gambar citra digital direpresentasikan dengan sebuah matriks  $n \times n$  dimana  $n$  sesuai dengan ukuran pikselnya. Misalkan apabila sebuah citra memiliki jumlah piksel  $50 \times 50$ , maka citra tersebut dapat direpresentasikan dengan matriks berukuran  $50 \times 50$ . Setiap elemen matriks merupakan bit-bit warna penyusun piksel tersebut. Contoh representasi citra digital dapat dilihat pada gambar 5.



**Gambar 5.** Representasi citra digital dengan matriks

Gambar 2.11 di atas merupakan ilustrasi citra digital hitam-putih dengan ukuran  $5 \times 5$  piksel. Didefinisikan bahwa bit “0” merepresentasikan piksel putih sedangkan bit “1” merepresentasikan piksel hitam. Untuk citra digital berwarna, masing-masing bit merepresentasikan setiap komponen warna pembentuk warna piksel.

### 2.2.2 Warna pada Citra Digital

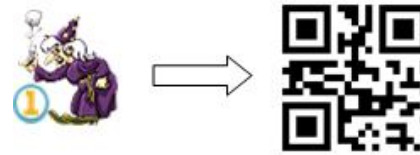
Ada beberapa jenis pewarnaan pada citra digital, yaitu warna hitam-putih (*black and white*), *grayscale*, dan citra berwarna. Citra hitam putih disebut juga citra satu bit, karena satu piksel hanya perlu direpresentasikan dengan satu bit data. Namun dari segi kualitas kurang baik karena hanya terdiri dari dua warna, hitam dan putih.

Pada pewarnaan *grayscale*, warna yang tersedia hanyalah warna diantara hitam dan putih. Oleh karena itu tidak terlalu banyak warna yang ditampilkan pada citra *grayscale*. Namun kualitas yang dihasilkan lebih baik daripada citra warna hitam putih. Pada pewarnaan *grayscale*, perubahan warna antara dua piksel yang berdekatan tidak terlihat signifikan sehingga gambar lebih mudah dicerna.

## 3. ANALISIS PENYELESAIAN MASALAH

### 3.1 Code Generator : Analisis Pemrosesan Image menjadi QR Code

*R Code* yang akan dikembangkan adalah *QR Code* yang terbentuk dari data berupa *image*, bukan teks seperti pada umumnya. Oleh karena itu masalah pertama yang harus diselesaikan adalah bagaimana merubah data berupa *image* menjadi sebuah *QR Code* seperti diilustrasikan pada gambar 6 di bawah ini.



**Gambar 6.** Ilustrasi pemrosesan *image* menjadi *QR Code*

Secara umum, terdapat dua permasalahan untuk melakukan pemrosesan *image* menjadi sebuah *QR Code*, yaitu :

1. Bagaimana mengubah *file image* menjadi representasi lain, misalnya data *byte*, numerik, atau alfanumerik.
2. Bagaimana mengubah data hasil representasi dari data *image* tersebut menjadi sebuah *QR Code*.

Agar dapat dibaca oleh sebuah *QR Code*, data berbentuk *image* harus diubah terlebih dahulu menjadi bentuk lain, misalnya *byte*, numerik, atau alfanumerik. Agar dapat dibaca oleh sebuah *QR Code*, data berbentuk *image* harus diubah terlebih dahulu menjadi bentuk lain, misalnya *byte*, numerik, atau alfanumerik. Selanjutnya bentuk lain tersebut dapat diubah dengan algoritma yang sudah tersedia.

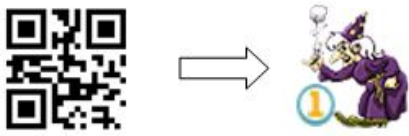
*QR Code* mengubah data dalam bit menjadi satu buah kotak, ‘0’ untuk hitam dan ‘1’ untuk putih. Oleh karena itu pada dasarnya semua tipe data, baik itu numerik, alfanumerik, atau tipe lainnya pada akhirnya akan diubah menjadi bit. Maka pada Tugas Akhir ini, *image* yang akan dijadikan *QR Code* direpresentasikan oleh data berupa *byte*, untuk kemudian setiap bit-nya akan langsung diproses. Hal ini lebih efisien daripada menggunakan tipe data lain sebagai perantara.

Namun pada kenyataannya, *QR Code generator* yang telah tersedia hanya menyediakan proses *encodation* untuk data bertipe *string*, bukan *byte*. Oleh karena itu pada Tugas Akhir ini, gambar masukan akan diubah menjadi representasi *bytestream* terlebih dahulu. *Bytestream* merupakan aliran bit dari *file* gambar masukan. *Bytestream* hasil pemrosesan *image* tersebut kemudian diubah menjadi teks alfanumerik. Teks alfanumerik inilah yang kemudian akan menjadi data masukan *QR Code generator* untuk diubah menjadi sebuah *QR Code*.

### 3.2 QR Code Reader : Analisis Pemrosesan QR Code menjadi Image

Pada Tugas Akhir ini, selain akan dibuat *QR Code generator* juga akan dibuat *QR Code reader*, yaitu aplikasi yang dapat membaca *QR Code* yang telah ada sehingga menjadi bentuk asalnya, dalam hal ini *image file* atau *file gambar*. Oleh karena itu masalah berikutnya adalah bagaimana memroses *QR Code* tersebut sehingga dapat menampilkan gambar seperti bentuk

sebelumnya atau singkatnya bagaimana men-*decode QR Code* seperti ilustrasi pada gambar 7.



**Gambar 7. Ilustrasi pemrosesan QR Code menjadi image**

Secara umum, terdapat dua permasalahan untuk melakukan pemrosesan *QR Code* menjadi sebuah *image*, yaitu :

1. Bagaimana menerjemahkan *QR Code* menjadi representasi data dari *file* gambar, misalnya berupa data *byte*, numerik, atau alfanumerik.
2. Bagaimana mengubah data hasil representasi dari *file* gambar tersebut menjadi sebuah gambar aslinya.

Selanjutnya untuk analisis *QR Code* reader merupakan reverse atau kebalikan dari analisis yang dilakukan untuk *QR Code* generator.

### 3.3 Analisis Kebutuhan Sistem

Kebutuhan fungsional sistem atau perangkat lunak yang dibangun adalah sebagai berikut :

1. Sistem dapat menerima masukan berupa *file* gambar, baik itu berbentuk *.jpeg*, *.jpg*, *.bmp*, *.gif*, maupun *.png*.
2. Sistem dapat mengubah *file* gambar masukan menjadi sebuah *QR Code* yang merepresentasikan *file image* tersebut.
3. Sistem dapat menyimpan gambar *QR Code* yang telah *generate* menjadi sebuah *file* gambar.
4. Sistem dapat menerima masukan berupa *file* gambar *QR Code* yang berasal dari *file* gambar.
5. Sistem dapat menerjemahkan *QR Code* dari *file* gambar menjadi gambar semula.
6. Sistem dapat menyimpan gambar hasil terjemahan dari *QR Code* menjadi sebuah *file* gambar.

## 4. PERANCANGAN DAN IMPLEMENTASI

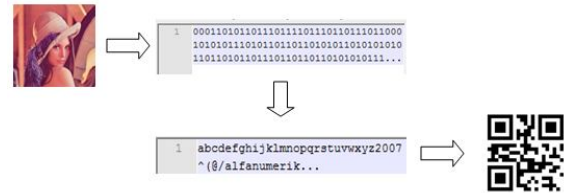
### 4.1 Deskripsi Umum Sistem

Perangkat lunak yang akan dibuat merupakan sebuah aplikasi berbasis *desktop* yang pada intinya memiliki dua fungsi, yaitu men-*generate* sebuah *QR Code* dari data berbentuk *image* (*QR Code generator*) serta membaca *QR Code* yang terbentuk dari data berbentuk *image* untuk mengembalikannya ke bentuk *image* tersebut (*QR Code reader*).

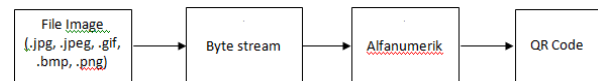
### 4.2 Deskripsi QR Code Generator

*QR Code generator* akan meminta masukan sebuah *file image* yang dapat diambil dari sistem operasi komputer. Kemudian *file* gambar tersebut dibaca sebagai *byte stream* sebelum diubah menjadi representasi *byte*, numerik, atau alfanumerik dengan algoritma tertentu untuk kemudian diubah menjadi *QR Code* dengan algoritma yang sudah tersedia. Berdasarkan analisis sebelumnya, maka representasi data yang digunakan adalah alfanumerik. Gambar *QR Code* yang terbentuk akan ditampilkan oleh perangkat lunak. *User* kemudian diberi pilihan untuk

menyimpan *QR Code* yang terbentuk menjadi sebuah *file* gambar serta untuk mencetak gambar tersebut. Ilustrasi dan diagram kerja perangkat lunak tersebut dapat dilihat pada gambar 8 dan 9 di bawah ini.



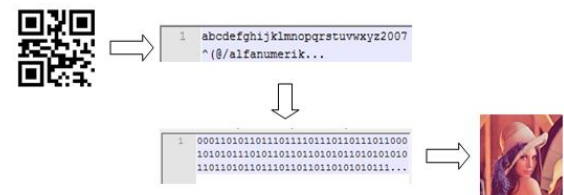
**Gambar 8. Ilustrasi proses kerja QR Code Generator dari data masukan image**



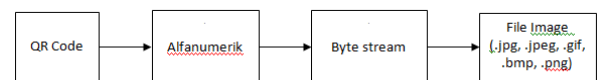
**Gambar 9. Diagram proses kerja QR Code Generator**

### 4.3 Deskripsi QR Code Reader

*QR Code reader* akan meminta masukan sebuah *file* gambar *QR Code* yang diambil dari sistem operasi komputer. Kemudian perangkat lunak ini akan membaca *QR Code* tersebut untuk kemudian menampilkan hasilnya kepada *user*, baik itu berupa teks maupun berupa gambar. Seperti halnya *QR Code generator*, pada *QR Code reader user* juga dapat menyimpan dan atau mencetak hasil pembacaan *QR Code*. Ilustrasi dan diagram kerja perangkat lunak tersebut dapat dilihat pada gambar 10 dan 11 di bawah ini.



**Gambar 10. Ilustrasi proses kerja QR Code Reader dari data berbentuk image**



**Gambar 11. Diagram proses kerja QR Code Reader**

### 4.4 Implementasi Perangkat Lunak

Implementasi perangkat lunak didasarkan pada analisis dan perancangan yang telah dilakukan. Implementasi yang dilakukan dapat dibagi menjadi dua bagian, yaitu implementasi kelas-kelas dan implementasi antarmuka. Implementasi dilakukan pada *notebook* dengan spesifikasi sebagai berikut :

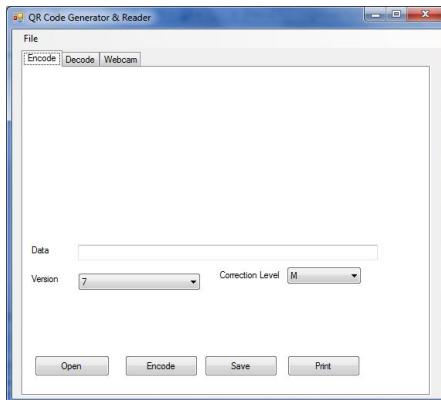
1. Hardware
  - a. CPU® Intel Core™ i3 350M (2,26 GHz)
  - b. RAM 2 GB
  - c. Harddisk 320 GB
  - d. Acer Crystal Eye Webcam Resolusi 640x480
2. Software
  - a. Sistem operasi Windows 7 Professional 32-bit

- b. Microsoft Visual Studio 2008
- c. .NET framework 3.5

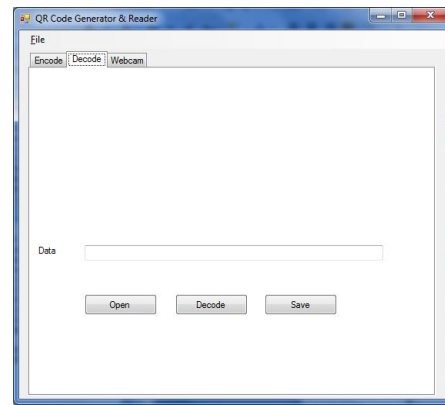
Bahasa pemrograman yang digunakan dalam mengimplementasikan perangkat lunak ini adalah C#. Pemilihan ini dilakukan karena *library* dan referensi lainnya, seperti *QR Code Generator* dan *QR Code Reader* yang sudah tersedia, telah banyak diimplmentasikan dalam bahasa C#. Oleh karena itu modifikasi yang dilakukan tidak memerlukan penyesuaian bahasa terlebih dahulu. *Tools* yang digunakan adalah Microsoft Visual Studio 2008.

Perangkat lunak terdiri dari 3 kelas utama, yaitu kelas *QRCodeSampleApplication*, kelas *Webcam*, dan kelas *Tools*. Selain kelas di atas, pada program ini juga terdapat *library* untuk melakukan proses *encoding* dan *decoding QR Code*. Kelas *QRCodeSampleApplication* merupakan kelas utama yang berisi *interface* serta *methods* yang berisi pemanggilan kelas lain dan *library QRCode*. Kelas *Webcam* adalah kelas yang berisi *methods* dan atribut untuk melakukan integrasi program dengan *webcam* pada *notebook*. Sedangkan kelas *Tools* berisi *methods converter* dan beberapa *method* bantuan lain.

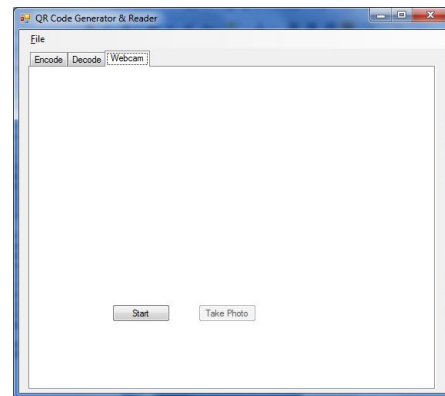
Perangkat lunak yang dibangun memiliki tiga tab utama, yaitu tab *QR Code generator*, tab *QR Code reader*, dan tab integrasi dengan *webcam*. Ketiga tab utama tersebut dapat dilihat pada gambar 12, 13, dan 14.



Gambar 12. Antarmuka tab *Encode*



Gambar 13. Antarmuka tab *Decode*



Gambar 14. Antarmuka tab *Webcam*

Implementasi perangkat lunak *QR Code generator* dan *QR Code reader* memiliki beberapa batasan, antara lain :

1. *QR Code generator* dan *QR Code reader* yang diimplementasikan adalah kode yang sudah tersedia di <http://www.codeproject.com/KB/cs/qrcode.aspx> untuk kemudian dimodifikasi agar masukan dan keluaran berupa *image*, bukan teks.
2. Kelas untuk *Webcam* juga diambil dari kode yang sudah tersedia di <http://www.c-sharpcorner.com/UploadFile/yougerthen/810262008070218AM/8.aspx> yang dimodifikasi sesuai dengan kebutuhan perangkat lunak.
3. Perangkat lunak dapat melakukan proses *decodation* dari foto *QR Code* yang dihasilkan *webcam*. Namun karena resolusi yang kurang baik, maka foto *QR Code* yang diambil terbatas ukurannya. Sistem tidak dapat men-*decode* foto *QR Code* dengan ukuran tertentu yang diambil dari *webcam*.
4. Ukuran *QR Code* ditentukan oleh *user*. Perangkat lunak tidak dapat secara otomatis menentukan ukuran *QR Code* minimum yang tepat untuk merepresentasikan sebuah *file* gambar.

## 5. PENGUJIAN

SAD Pengujian yang dilakukan terhadap perangkat lunak yang dibangun pada Tugas Akhir ini pada intinya bertujuan

untuk mengetahui kemungkinan membangkitkan *QR Code* dari data berbentuk *image*. Pada pengujian ini diharapkan dapat menemukan kesimpulan apakah perangkat lunak *QR Code generator* dan *QR Code reader* dari data berbentuk *image* ini *feasible* untuk diterapkan di dunia nyata. Secara rinci, tujuan pengujian perangkat lunak ini antara lain :

1. mengukur fungsionalitas dari perangkat lunak. Pengujian ini mengacu pada analisis kebutuhan sistem yang telah dijabarkan sebelumnya.
2. mengukur performansi dari perangkat lunak. Pengujian ini bertujuan untuk mengukur perbandingan kualitas gambar asli dengan gambar hasil pembacaan *QR Code*. Selain itu, tujuan lainnya adalah untuk mengetahui ukuran maksimum dari *file* gambar yang dapat dijadikan *QR Code*.

Dalam pengujian aplikasi *QR Code generator* dan *QR Code reader* ini, terdapat beberapa skenario pengujian yang dapat dibagi menjadi dua bagian, yaitu pengujian fungsionalitas dan pengujian performansi.

Pengujian fungsionalitas ini, seperti telah dikatakan sebelumnya, mengacu pada analisis kebutuhan perangkat lunak yang telah dijabarkan sebelumnya. Pada intinya, yang dilakukan pada tahap ini adalah menguji apakah sistem dapat bekerja seperti yang diharapkan pada analisis kebutuhan.

Pengujian performansi dilakukan untuk membandingkan kualitas gambar sebelum diubah menjadi *QR Code* dengan gambar setelah *QR Code* tersebut diterjemahkan menjadi *file* gambar kembali. Pada pengujian ini, perangkat lunak akan diberi masukkan beberapa *file* gambar dengan tipe dan ukuran yang bervariasi, untuk selanjutnya *file* gambar tersebut diubah menjadi *QR Code* oleh *QR Code generator*. Kemudian *QR Code* yang terbentuk langsung dibaca oleh *QR Code reader* untuk mengembalikannya ke bentuk *file* gambar. Setelah itu dilakukan perbandingan antara gambar semula dengan gambar hasil pembacaan *QR Code*, yang meliputi perbandingan resolusi dan ukuran *file*.

Pengujian performansi berikutnya adalah menentukan ukuran maksimum *file* gambar yang dapat dijadikan *QR Code*, yaitu dengan cara memasukkan gambar-gambar dengan berbagai ukuran untuk setiap versi *QR Code*. Dengan demikian akan didapat ukuran maksimum gambar yang dapat disimpan pada setiap versi *QR Code*. Gambar yang dimasukkan adalah gambar *apply.gif* yang dimodifikasi menjadi berbagai ukuran.

Kemudian pengujian performansi selanjutnya adalah mengukur pengaruh dari *noise* yang ada pada *QR Code*, terutama pada *QR Code* hasil dari foto kamera, terhadap keterbacaan data yang disimpan pada *QR Code*. Pada pengujian ini akan dilakukan pembacaan beberapa *file QR Code* yang kurang sempurna, baik itu karena buram, permukaan bergelombang, dan *noise* lainnya.

Hasil pengujian fungsional dan pengujian performansi dapat dilihat pada tabel berikut ini. Tabel 1 menunjukkan hasil pengujian fungsionalitas perangkat lunak beserta status keberhasilannya dibandingkan dengan analisis kebutuhan sistem yang telah dilakukan. Tabel 2 menggambarkan ukuran maksimum *file* gambar yang dapat ditampung oleh *QR Code* per

versi. Sedangkan tabel 3 merupakan perbandingan kualitas gambar sebelum dan setelah pemrosesan menjadi *QR Code*.


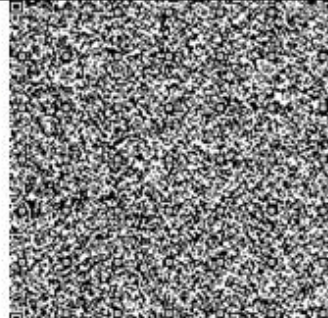


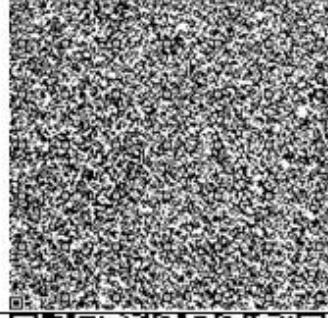




**Tabel 1. Hasil Pengujian Fungsionalitas**

| Pengujian  | Kriteria Keberhasilan   | Status   |
|--|---|----------|
| Menerima masukan <i>file</i> gambar                      | Perangkat lunak dapat menerima masukan berupa <i>file</i> gambar dengan berbagai kemungkinan tipe <i>file</i> . <i>File</i> yang bukan gambar akan ditolak. | Berhasil |
| Membangkitkan <i>QR Code</i> dari <i>file</i> gambar     | Perangkat lunak dapat mengubah <i>file</i> gambar masukan menjadi sebuah <i>QR Code</i> dengan ukuran tertentu.   | Berhasil |
| Menyimpan <i>file</i> gambar <i>QR Code</i>              | Perangkat lunak dapat menyimpan <i>QR Code</i> yang telah dibangkitkan menjadi sebuah <i>file</i> gambar  | Berhasil |
| Menerima masukan <i>file</i> gambar <i>QR Code</i>       | Perangkat lunak dapat menerima masukan berupa <i>file</i> gambar <i>QR Code</i> dengan berbagai kemungkinan tipe <i>file</i> .                              | Berhasil |
| Membaca <i>QR Code</i> menjadi <i>file</i> gambar semula | Perangkat lunak dapat membaca <i>QR Code</i> masukan dan menerjemahkannya menjadi sebuah gambar.  | Berhasil |
| Menyimpan <i>file</i> gambar keluaran                    | Perangkat lunak dapat menyimpan gambar hasil terjemahan <i>QR Code</i> menjadi sebuah <i>file</i> gambar.   | Berhasil |

**Tabel 2. Ukuran maksimum gambar per versi QR Code**

| Versi <i>QR Code</i> | Ukuran maksimum gambar |              |
|----------------------|------------------------|--------------|
|                      | <i>Pixel</i>           | <i>Bytes</i> |
| 1                    | -                      | -            |
| 6                    | -                      | -            |
| 11                   | 5 x 5                  | 153 bytes    |
| 16                   | 8 x 8                  | 285 bytes    |
| 21                   | 8 x 8                  | 285 bytes    |
| 26                   | 12 x 12                | 552 bytes    |
| 31                   | 26 x 26                | 1.18 kB      |
| 36                   | 38 x 38                | 1.53 kB      |
| 40                   | 48 x 48                | 1.86 kB      |

Tabel 3. Perbandingan Gambar Hasil Pengujian

| Gambar Awal  | Gambar QR Code   | Gambar hasil Pembacaan QR Code   |
|--|--|--|
| <br>blue.png<br>Ukuran : 1.53 kb  |   | <br>blue.png<br>Ukuran : 1.16 kb  |
| <br>apply.gif<br>Ukuran : 1.51 kb |   | <br>apply.gif<br>Ukuran : 1.82 kb |
| <br>as.gif<br>Ukuran : 172 bytes |  | <br>as.gif<br>Ukuran : 226 bytes |

Analisis hasil pengujian dari implementasi *QR Code generator* dan *QR Code reader* dari data masukan berbentuk *image* yang dibuat pada Tugas Akhir ini meliputi beberapa aspek sesuai yang telah dijabarkan pada tujuan pengujian. Aspek-aspek yang dimaksud adalah aspek fungsionalitas dan aspek performansi dari perangkat lunak yang dibangun.

Pengujian dari aspek fungsionalitas hampir sesuai dengan analisis kebutuhan sistem. Perangkat lunak dapat menerima masukan berupa *file gambar* dan hanya *file gambar*, dimana tipe *file* lain tidak bisa diakses untuk menjadi masukan. Setelah itu *file gambar* dapat dibaca oleh *QR Code generator* dan diubah menjadi sebuah *QR Code*. Kemudian perangkat lunak ini dapat menyimpan *file gambar QR Code* tersebut jika diinginkan. Selanjutnya, perangkat lunak juga dapat memanggil kembali *file gambar QR Code* untuk dibaca oleh *QR Code reader* menjadi sebuah *file gambar* aslinya. Gambar hasil terjemahan *QR Code* ini pun dapat disimpan menjadi sebuah *file* bertipe gambar.

Pengujian dari aspek performansi perangkat lunak ini jika dilihat secara sepiantas secara visual tidak ada perubahan sama sekali. Dengan kata lain gambar sebelum proses *encodation-decodation*

dan setelah proses *encodation-decodation* sama. Namun jika dilihat pada *properties file* gambar, dapat diketahui bahwa terdapat sedikit perubahan ukuran pada *file gambar* tersebut, baik bertambah maupun berkurang. Salah satu contohnya pada gambar *apply.gif*, ukuran semula adalah 1.51 *kilobytes*, namun setelah melalui proses *encodation* dan *decodation*, ukuran *file gambar* berubah menjadi 1.82 *kilobytes*.

Ukuran maksimum *file gambar* yang dapat dijadikan *QR Code* adalah 1.86 *kilobytes* dengan ukuran 48 x 48 pixel. Oleh karena itu, hanya gambar-gambar kecil berupa *icon* atau lambang sederhana saja yang dapat dijadikan *QR Code* berdasarkan kapasitas *QR Code* yang ada saat ini.

Pengujian performansi untuk *webcam* dinilai kurang berhasil memenuhi ekspektasi. Hal ini karena untuk gambar diperlukan ukuran *QR Code* yang besar, yang berarti diperlukan resolusi kamera yang jauh lebih baik untuk membaca setiap sel data yang relatif kecil.

## 6. KESIMPULAN

Setelah melakukan studi literatur, analisis, perancangan, implementasi, dan diakhiri dengan pengujian perangkat lunak yang dibangun, maka dapat diambil beberapa kesimpulan berikut :

1. *QR Code* dari data berbentuk *image* dapat dibuat, namun tidak *feasible* untuk diterapkan di dunia nyata. Hal ini karena *QR Code* yang dibangun dari data gambar sangat besar dan sulit dibaca. Selain itu hanya gambar dengan ukuran kecil yang dapat dibentuk menjadi *QR Code*.
2. *QR Code* dari data berbentuk *image* dapat dibuat dengan cara dilakukan pemrosesan *file image* menjadi *byte stream* terlebih dahulu, kemudian *byte stream* tersebut diproses menjadi alfanumerik sebagai representasi data yang memungkinkan untuk diproses kedalam *QR Code generator*. Begitu pula sebaliknya untuk memroses *QR Code* menjadi *image* kembali.
3. Ukuran maksimum *file gambar* yang dijadikan *QR Code* tergantung dari versi *QR Code*. Untuk versi terbesar, yaitu versi 40, ukuran maksimum *file gambar* masukkan adalah sebesar 1.86 *kilobytes* dengan ukuran 48 x 48 pixel.

## 7. ACKNOWLEDGMENTS

Alhamdulillah penulis panjatkan kehadiran Allah SWT atas dibuatnya *paper* ini. Penulis juga ingin mengucapkan terima kasih kepada berbagai pihak yang telah membantu penulis dalam penulisan *paper* ini. Terima kasih juga penulis sampaikan kepada ACM SIGCHI yang telah menyediakan *template* penulisan *paper* untuk penulis modifikasi.

## 8. REFERENCES

- [1] Adha, Ripandy. 2010. *Message Digest dalam bentuk QR Code Sebagai Tanda Tangan Digital*. Bandung: Program Studi Teknik Informatika STEI ITB.
- [2] Bejaoui, Bechir. 2008. *Integrate the web Webcam functionality using C#.Net and COM part VIII*. Diakses

- pada 20 Juli 2011 di <http://www.c-sharpcorner.com/UploadFile/yougerthen/810262008070218AM/8.aspx>
- [3] Denso Wave. 2010. *QR Code Introduction – Symbol Version*. Diakses pada 25 Agustus 2011 di <http://www.denso-wave.com/QRCode/QRgene2-e.html>
- [4] Firdaus. 2010. *Dampak Teknologi Informasi dan Komunikasi Dalam Dunia Pendidikan*. Diakses Desember 2010 di <http://firdaus666.blog.upi.edu/2010/11/07/makalah-dampak-teknologi-informasi-dan-komunikasi-dalam-dunia-pendidikan-2/>
- [5] ISO/IEC 18004. 2000. *Information Technology – Automatic Identification and Data Capture Techniques – Bar Code Symbolology – QR Code*. Swizerland : International Standard
- [6] ITSC. 2008. *Section 3: QR Code*. Synthesis Journal. Information Technology Standards Committee Singapore. Diakses pada tanggal 17 November 2010, dari [http://www.itsc.org.sg/pdf/synthesis08/Three\\_QR\\_Code.pdf](http://www.itsc.org.sg/pdf/synthesis08/Three_QR_Code.pdf).
- [7] Kuan, Chieh Liao & Hsun Lee Wei. 2010. *A Novel User Authentication Scheme Based on QR-Code*. Taichung, Taiwan: Journal of Networks, Vol.5, No.8. pp937-941.
- [8] Law, Chin-Ying & Simon So. 2007. *QR Codes in Education*. Hongkong: Journal of Educational Technology Development and Exchange, 3(1), 85-100.
- [9] Liu, Yue & Mingjun Liu. 2006. *Automatic Recognition Algorithm of Quick Response Code Based on Embedded System*. Jinan, China: Proceedings of the Sixth International Conference on Intelligent System Design and Applications.
- [10] Mihardja, Taufik. 2009 *QR Code Kompas Perkaya Konten bagi Pembaca*. Diakses pada Desember 2010 di <http://tekno.kompas.com/read/2009/06/15/0850503/QR.Cod.e.Kompas.Perkaya.Konten.bagi.Pembaca>
- [11] Paryono, Petrus, dkk. 2005. *Citra Digital*. Yogyakarta : Universitas Kristen Duta Wacana.
- [12] Rahayu, Yeni Dwi, dkk. 2006. *Pembuatan Aplikasi Pembacaan Quick Response Code Menggunakan Perangkat Mobile Berbasis J2ME Untuk Identifikasi Suatu Barang*. Surabaya: Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember.
- [13] Rulez, Shinigami. 2011. *Pengolahan Citra Digital* diambil dari <http://shinigamirulez.blogspot.com/2009/01/pengolahan-citra-digital.html> diakses pada 30 Juni 2011
- [14] Suryadi, Dedi. *Teknologi Informasi dalam Sistem Informasi Manajemen (SIM) Pendidikan Menengah Kejuruan*. Bandung: FTPK UPI.
- [15] Twit88. 2007. *Open Source QRCode Library*. Diakses pada November 2011 di <http://www.Codeproject.com/KB/cs/QRCode.aspx>
- [16] Pusat Barcode. 2008. *Definisi dan Sejarah Barcode*. Diakses pada Desember 2010 di <http://pusatbarcode.wordpress.com/2008/10/27/definisi-dan-sejarah-barcode/>