

# Pencarian Informasi dengan Pencocokan Pola pada QR Code

Holy Lovenia / 13515113  
Program Studi Teknik Informatika  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia  
13515113@std.stei.itb.ac.id  
holy.lovenia@gmail.com

**Abstrak**—QR (Quick Response) code merupakan sebuah barcode dua dimensi yang berbentuk matriks. QR code terdiri dari persegi-persegi hitam yang diatur pada kisi-kisi dengan latar belakang putih. Pada umumnya, QR code memuat informasi dalam bentuk data optik spesifik yang bisa diperoleh melalui pembacaan mesin dari komponen-komponen vertikal dan horizontalnya. Empat mode pengkodean (termasuk ekstensi) yang digunakan oleh QR code dalam penyimpanan data adalah numerik, alfanumerik, byte / biner, dan kanji. Dalam hal pembacaan QR code, diterapkan pencocokan pola seperti algoritma Knuth-Morris-Pratt.

**Kata kunci**—QR code; basis data; informasi; Knuth-Morris-Pratt; pencarian; pencocokan; pola; string.

## I. PENDAHULUAN

Pada awalnya, QR code digunakan dalam industri otomotif untuk membantu pelacakan kendaraan atau komponen-komponennya pada proses manufaktur. QR code yang merupakan pengembangan dari barcode didesain oleh anak perusahaan Toyota, yaitu Denso Wave, sedemikian rupa sehingga memiliki kecepatan dekoding yang lebih tinggi dibanding barcode biasa. Karena kecepatan, akurasi, dan multifungsionalitas yang tersedia, pemanfaatan QR code mulai marak di berbagai sektor kehidupan masyarakat. [1]

Setelah itu, banyak pengembangan dan usaha yang dilakukan untuk memperbaiki teknologi QR code yang sudah ada dan memperbesar kapasitas informasi yang dapat ditampung. Beberapa usaha di antaranya adalah meningkatkan jumlah digit dalam kode, perbaruan tata letak agar bisa menyisipkan lebih banyak kode, dan modifikasi fungsionalitas. Hal ini ditanggulangi dengan lahirnya QR code dengan bentuk dua dimensi seperti sekarang.

Dewasa ini, sistem QR code digunakan dalam konteks yang jauh lebih luas, contohnya pada pengenalan aplikasi secara komersial atau aplikasi-aplikasi mobile yang berbasis kenyamanan pengguna. QR code dapat dimanfaatkan untuk menampilkan teks tertentu, menambahkan kontak vCard pada alat komunikasi

pengguna, membuka Uniform Resource Identifier (URI) seperti Uniform Resource Locator (URL) atau lebih dikenal sebagai tautan / alamat web, maupun mengirim pesan elektronik.

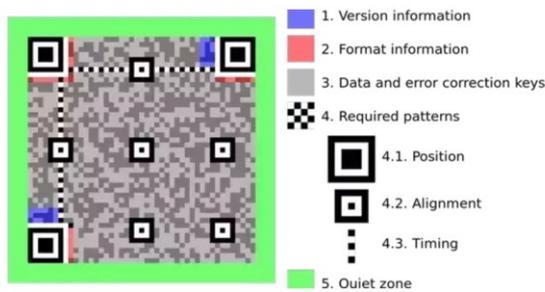
QR code kini umum digunakan untuk iklan-iklan produk atau jasa secara komersial oleh berbagai industri. Konsumen dapat memanfaatkan telepon pintar yang dimiliki sebagai pemindai QR code untuk menampilkan kode dan mengonversikannya ke bentuk yang lebih berguna, seperti URL menuju web produk atau jasa tersebut. Beberapa aktivitas lainnya yang dapat melibatkan pemanfaatan QR code adalah pemeriksaan tiket pada transportasi ataupun acara-acara hiburan, pemasaran produk dan jasa (seperti kupon mobile atau potongan harga), publikasi informasi perusahaan seperti alamat atau keterangan alfanumerik lainnya, label produk dalam toko, dan lainnya. QR code juga dapat dikaitkan dengan geolokasi melalui Global Positioning System (GPS), kurs kriptografik seperti Bitcoin, log in dalam situs tertentu, maupun identitas-identitas pribadi seperti informasi akun bank atau kartu kredit.

Dengan berbagai fungsionalitasnya, QR code juga dapat diposisikan sebagai primary key dalam sebuah basis data, salah satunya terkait media sosial. Pada dewasa ini, seringkali pencarian pengguna pada media sosial dilakukan dengan cara memindai QR code yang bersangkutan. Hasil pindai QR code diterima dalam bentuk teks. Untuk mendapatkan informasi pengguna yang dicari, sistem perlu melakukan pencocokan kode hasil pindai dengan informasi-informasi yang ada dalam basis data. Beberapa algoritma yang dapat dipakai untuk melakukan pencocokan pola dalam bentuk teks adalah bruteforce, Knuth-Morris-Pratt, dan Boyer-Moore.

## II. DASAR TEORI

### A. QR Code

QR code dibedakan berdasarkan data yang terkandung di dalamnya. Selain itu, ada beberapa cara untuk mengidentifikasi QR code. [2]



Gambar 1 Struktur QR Code (sumber: [https://en.wikipedia.org/wiki/QR\\_code#Standards](https://en.wikipedia.org/wiki/QR_code#Standards))

1. *Version* dan *format*: digunakan agar alat pemindai mengenali jenis data yang terkandung di dalam QR code.
2. Data dan *error correction*: walaupun data sedikit kabur atau hilang, terkadang QR code yang menggunakan tingkat koreksi kesalahan tertentu pada saat penulisan kode, tetap dapat terbaca.
3. *Position*: digunakan untuk mendeteksi posisi dan mengenali sudut-sudut dari QR code.
4. *Alignment*: digunakan untuk membantu navigasi titik referensi pada kode-kode yang lebih besar.
5. *Timing*: digunakan untuk menentukan besar dari modul juga posisi dari baris dan kolom.
6. *Quiet zone*: membutuhkan pinggirannya senilai sedikitnya empat modul.

Selain itu, berdasarkan data yang bisa dimasukkan oleh pengguna, ada empat mode QR code sebagai berikut.

1. Numerik: digit 0-9.
2. Alfanumerik: digit 0-9, huruf kapital (tidak termasuk huruf *lowercase*), simbol-simbol (\$, %, \*, +, -, ,, /, and :), dan spasi.
3. Byte: secara standar, digunakan untuk karakter dari ISO-8859-1. Walaupun begitu, beberapa pemindai QR code, dapat digunakan untuk mendeteksi UTF-8.
4. Kanji: digunakan untuk karakter *double-byte* dari karakter set Shift JIS, mengkode dan mendukung karakter bahasa Jepang.

Kata-kata kode pada QR code memiliki panjang 8 bit dan menggunakan algoritma Reed-Solomon *error correction* dengan empat tingkat perbaikan kesalahan.

1. Level L (Low): 7% dari kata-kata kode dapat dikembalikan.
2. Level M (Medium): 15% dari kata-kata kode dapat dikembalikan.
3. Level Q (Quartile): 25% dari kata-kata kode dapat dikembalikan.
4. Level H (High): 30% dari kata-kata kode dapat dikembalikan.

Pesan dipecah menjadi beberapa blok kode Reed-Solomon. Ukuran blok disesuaikan agar maksimum 15 kesalahan yang perlu diperbaiki pada setiap blok dan membatasi kompleksitas dari algoritma yang digunakan untuk dekoding. Karena toleransi yang diberikan oleh koreksi kesalahan, QR code dapat dipindai dengan benar sekaligus memuat warna, logo, dan beberapa fitur.

## B. Basis Data

Basis data adalah kumpulan data yang terorganisasi. Pada umumnya, data diatur berdasarkan aspek model dari realita untuk mendukung proses yang membutuhkan informasi. Berikut ini adalah empat fungsi utama pada sistem manajemen basis data. [3]

1. Definisi data: menciptakan, memodifikasi, dan menghapus deskripsi yang mendefinisikan organisasi data.
2. Memperbarui data: menambahkan, memodifikasi, dan menghapus data sebenarnya.
3. Mengambil data: menyediakan informasi dalam suatu bentuk yang dapat digunakan atau diproses lebih jauh oleh aplikasi lainnya.
4. Administrasi data: mendaftarkan dan memonitor pengguna, menjaga keamanan data, mengamati performa, menjaga integritas data, menangani konkurensi, dan mengembalikan informasi yang rusak karena kegagalan sistem atau kejadian tak terduga lainnya.

Basis data relasional menyimpan informasi tentang relasi antar entitas (benda yang simpan keterangannya). Setiap entitas pada basis data memiliki *primary key*, yaitu atribut kunci yang memiliki nilai unik untuk jenis entitas tersebut sebagai identitas entitas.

## C. Pencocokan Pola

Pencocokan *string* (*pattern matching*) merupakan pencarian *string* di dalam teks. Persoalan pencarian *string* memiliki teks (*text*), yaitu *string* yang memiliki panjang  $n$  karakter, dan pola (*pattern*) yang merupakan *string* dengan panjang  $m$  karakter ( $m < n$ ) yang akan dicari pada teks. *Pattern matching* akan mencari posisi pertama yang sesuai dengan pola di dalam teks. [4]

Sepotong bagian dari teks dapat berupa sufiks atau prefiks. Pada *string*  $S$  yang memiliki panjang  $m$ , prefiks dari  $S$  adalah bagian dari *string* yang berada di antara posisi ke-0 hingga  $k$  dan sufiks dari  $S$  merupakan bagian dari *string* yang berada di antara posisi  $k$ , di mana  $k$  adalah indeks yang berada di antara 0 hingga  $m-1$ .

### 1. Algoritma Bruteforce

Pada pencocokan *string*, algoritma *bruteforce* melakukan pencarian *string* dengan melakukan perbandingan pada setiap karakter yang dijumpai hingga menemukan bagian dari teks yang sesuai

dengan pola. Bila teks berada dalam tabel karakter  $T[0..n]$  dan pola berada dalam tabel karakter  $P[0..m]$ , maka proses pencocokan *string* dengan algoritma *bruteforce* adalah sebagai berikut.

1. Indeks P ( $idxP$ ) dimulai dari 0 dan indeks T ( $idxT$ ) dimulai dari 0.
2. Karakter P yang berada pada posisi  $idxP$  dibandingkan dengan karakter T yang berada pada posisi  $idxT$ .
3. Jika  $P[idxP]$  dan  $T[idxT]$  cocok, maka pencarian akan dilanjutkan dengan  $idxP$  dan  $idxT$  berikutnya hingga semua karakter pada P cocok. Sedangkan apabila  $P[idxP]$  dan  $T[idxT]$  tidak cocok dan teks T belum mencapai akhir, pada pencarian selanjutnya  $idxP$  akan dimulai dari 0 kembali dan  $idxT$  maju ke  $idxT$  berikutnya.

Pada algoritma *bruteforce*, kasus terbaik yang dapat terjadi adalah ketika karakter pertama pada pola tidak pernah sama dengan karakter teks T yang dicocokkan. Jumlah perbandingan yang dilakukan maksimal  $n$  kali, sehingga kompleksitas kasus terbaiknya adalah  $O(n)$ . Sedangkan, kasus terburuknya adalah ketika indeks prefiks terpanjang dari pola cocok dengan setiap prefiks teks yang panjangnya lebih dari atau sama dengan panjang pola, tetapi baru menemukan kecocokan pola pada akhir teks. Jumlah perbandingan yang dilakukan adalah  $m(n-m+1)$  kali, sehingga kompleksitas waktu terburuknya adalah  $O(mn)$ .

## 2. Algoritma Knuth-Morris-Pratt (KMP)

Pada algoritma KMP, pencocokan *string* juga dilakukan dengan urutan dari kiri ke kanan seperti algoritma *bruteforce*. Perbedaannya terletak dari pergeseran pola yang dilakukan ketika melakukan pencocokan. Ketika ketidakcocokan pola dan teks muncul, sistem akan menggeser pola sejauh mungkin agar tidak melakukan perbandingan berlebihan. Hal ini dilakukan dengan memanfaatkan proses prakomputasi pola untuk menemukan kecocokan prefiks pola dengan pola itu sendiri.

Prakomputasi pola pada algoritma KMP memanfaatkan fungsi pinggiran (*border function*), yaitu panjang maksimum dari prefiks pola yang juga merupakan sufiks dari pola yang memiliki panjang  $j$ . Contoh penggunaan *border function* dengan pola  $P = \text{"ababababca"}$  adalah sebagai berikut.

Tabel 1 Fungsi Pinggiran

$J$	0	1	2	3	4	5	6	7	8	9
$P[j]$	a	b	a	b	a	b	a	b	c	a
$b[j]$	0	0	1	2	3	4	5	6	0	1

Kompleksitas waktu algoritma KMP adalah  $O(m+n)$ , karena untuk menghitung fungsi pinggiran dibutuhkan waktu sebesar  $O(m)$  dan pencarian *string* sebesar  $O(n)$ .

Algoritma untuk menghitung fungsi pinggiran adalah sebagai berikut.

```

procedure HitungPinggiran(input m:
integer, P: array[1..m] of char,
output b: array[1..m] of integer)
{ Menghitung nilai b[1..m] untuk
pattern P[1..m] }

Deklarasi
k,q: integer

Algoritma
b[1] ← 0
q ← 2
k ← 0
for q ← 2 to m do
  while ((k > 0) and (P[q] ≠ P[k+1]))
  do
    k ← b[k]
  endwhile
  if (P[q] = P[k+1]) then
    k ← k+1
  endif
  b[q] = k
endfor

```

## 3. Algoritma Boyer-Moore

Pencocokan *string* dengan algoritma Boyer-Moore menggunakan dua teknik, yaitu *looking-glass* untuk mencari pola P pada teks T dengan mengeceknya dari akhir ke awal, dan teknik *character-jump* yang dilakukan ketika terjadi ketidakcocokan pada  $T[i] = P[j]$ . Berikut ini adalah 3 kasus ketidakcocokan yang dapat terjadi.

1. Apabila  $T[i]$  berisi elemen  $x$  dan pola P mengandung elemen  $x$  di mana posisinya berada sebelum  $P[j]$ , maka pola P akan bergeser ke kanan untuk menyejajarkan posisi kemunculan terakhir  $x$  pada P dan  $T[i]$ , lalu kembali memulai perbandingan dari kanan.
2. Apabila  $T[i]$  berisi elemen  $x$  dan pola P mengandung elemen  $x$  di mana posisinya berada setelah  $P[j]$ , maka pola P akan bergeser satu karakter agar sejajar dengan  $T[i+1]$ .
3. Pada kasus yang tidak tercakup dalam kasus 1 dan 2, pola P bergeser untuk menyejajarkan  $P[0]$  dengan  $T[i+1]$ .

Algoritma Boyer-Moore melakukan prakomputasi pola P dan variasi dari setiap karakter yang ada di dalam teks T untuk membentuk fungsi kemunculan terakhir (*last occurrence function*). Berikut ini adalah contoh *last occurrence function* pada algoritma Boyer-Moore dengan pola  $P = \text{"abacab"}$  dan teks T yang mengandung huruf a, b, c, dan d.

Tabel 2 Kemunculan Terakhir

<b>x</b>	<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>
<b>L(x)</b>	4	5	3	-1

### III. STUDI KASUS: PEMANFAATAN PENCOCOKAN POLA PADA QR CODE DALAM PENCARIAN INFORMASI PENGGUNA MEDIA SOSIAL

#### A. Gambaran Umum

Pada persoalan berikut, sistem memiliki basis data yang berisi identitas-identitas pengguna pada sebuah media sosial. Informasi pengguna yang ada dalam basis data antara lain adalah *usercode*, *username*, nama depan, nama belakang, nomor telepon, dan *e-mail*. *Usercode* merupakan *primary key* dari tabel relasi tersebut dan adalah representasi dari QR code berbeda yang dimiliki oleh setiap pengguna.

Seorang pengguna dapat mencari informasi tentang pengguna lainnya dengan cara memindai QR code milik pengguna yang ingin dicari tersebut. Sistem akan memindai QR code yang diminta dan mendapatkan sebuah teks berisi *usercode* pengguna tersebut. Kemudian, sistem akan melakukan pencocokan pola antara *usercode* yang diperoleh dengan informasi *usercode-usercode* yang berada dalam basis data.

Tabel 3 Basis Data Sosial Media

Usercode	Username	First Name	Last Name	E-mail	Phone
12345	alibaba	Ali	Baba	alibaba@ggmai.cco	081202352856
12346	cliffbrown	Cliff	Brown	cliff_brown@harvestmoon.com	098777550234
12347	roxannecontraire	Roxanne	Contraire	roxanne@rocketmail.com	022777155156
12348	clairerivers	Claire	Rivers	claire_rivers@gmail.com	089955112323
12349	maryhadalittlelamb	Mary	Little	mary_had_a_little_lamb@gmail.com	0878714123123
12350	stuartjackson	Stuart	Jackson	stuartjackson@ymail.com	022456928380
12351	grayblacksmith	Gray	Blacksmith	gray.blacksmith@gmail.com	0871234532609
12352	doctortrent	Doctor	Trent	doctortrent@yahoo.com	08787878787
12353	anntheinn	Anna	Theinn	anntheinn@gmail.com	6571928371
12354	pinkpopuri	Pink	Popuri	pink_popuri@msn.co.id	0811234234234

#### B. Dekode QR Code

Ada beberapa pilihan cara untuk mendapatkan *usercode* dari hasil pemindaian QR code.

1. Menggunakan aplikasi QR code scanner pada telepon pintar.
2. Memakai QR code scanner yang tersedia pada jejaring online.
3. Membangun kode program untuk membaca QR code secara mandiri dengan library Java untuk Android yang tersedia seperti Zebra Crossing (ZXing), Android Mobile Vision API (Application Program Interface), dan lainnya. [5]

Setelah QR code berhasil dipindai, sistem akan memiliki *usercode* dari pengguna yang ingin dicari.

#### C. Pencocokan Pola untuk Pencarian Usercode

Pencocokan pola dapat dilakukan dengan tiga jenis algoritma, yaitu *bruteforce*, Knuth-Morris-Pratt, dan Boyer-Moore. Pada kasus ini, *usercode* yang dicari adalah "12353". Berikut ini adalah proses pencocokan yang dilakukan oleh setiap algoritma dengan *usercode* setiap pengguna pada basis data.

##### 1. Bruteforce

```

Pencocokan usercode ke-1
12345
1235

Pencocokan usercode ke-2
12346
1235

Pencocokan usercode ke-3
12347
1235

Pencocokan usercode ke-4
12348
1235

Pencocokan usercode ke-5
12349
1235

Pencocokan usercode ke-6
12350
12353

Pencocokan usercode ke-7
12351
12353

Pencocokan usercode ke-8
12352
12353

Pencocokan usercode ke-9
12353
12353
    
```

Gambar 2 Pencocokan Pola dengan Bruteforce

## 2. Knuth-Morris-Pratt

```
Pencocokan usercode ke-1
12345
1235
  1
  1

Pencocokan usercode ke-2
12346
1235
  1
  1

Pencocokan usercode ke-3
12347
1235
  1
  1

Pencocokan usercode ke-4
12348
1235
  1
  1

Pencocokan usercode ke-5
12349
1235
  1
  1

Pencocokan usercode ke-6
12350
12353
  1

Pencocokan usercode ke-7
12351
12353
  1

Pencocokan usercode ke-8
12352
12353
  1

Pencocokan usercode ke-9
12353
12353
  1
```

Gambar 3 Pencocokan Pola dengan Knuth-Morris-Pratt

## 3. Boyer-Moore

```
Pencocokan usercode ke-1
12345
  3

Pencocokan usercode ke-2
12346
  3

Pencocokan usercode ke-3
12347
  3

Pencocokan usercode ke-4
12348
  3

Pencocokan usercode ke-5
12349
  3

Pencocokan usercode ke-6
12350
  3

Pencocokan usercode ke-7
12351
  3
```

```
Pencocokan usercode ke-8
12352
  3

Pencocokan usercode ke-9
12353
  3
  53
  353
  2353
12353
```

Gambar 4 Pencocokan Pola dengan Boyer-Moore

Usercode “12353” ditemukan di basis data pada informasi dengan indeks ke-9.

### D. Pengambilan Data Berdasarkan Usercode

Setelah melakukan pencocokan pola, sistem akan menerima informasi bahwa *usercode* ditemukan pada indeks tertentu (dalam kasus ini indeks adalah 9) di basis data atau -1 apabila *usercode* tidak ada terdaftar pada media sosial tersebut. Kemudian, sistem akan mengambil data pengguna pada indeks tersebut dan menampilkannya ke layar pengguna media sosial yang memindai QR *code*.

```
QR Code: 12353
Username: anntheinn
First Name: Annabeth
Last Name: Theinn
E-mail: anntheinn@gmail.com
Phone: 6571928371
```

Gambar 5 Hasil Pengambilan Data

## IV. PERBANDINGAN ALGORITMA PENCOCOKAN POLA

Berdasarkan pencocokan pola yang dilakukan, algoritma *bruteforce*, Knuth-Morris-Pratt, dan Boyer-Moore memiliki waktu eksekusi yang berbeda sebagai berikut.

### 1. *Bruteforce*

Waktu eksekusi: 14573 ms

Gambar 6 Waktu Eksekusi *Bruteforce*

### 2. Knuth-Morris-Pratt

Waktu eksekusi: 6329 ms

Gambar 7 Waktu Eksekusi Knuth-Morris-Pratt

### 3. Boyer-Moore

Waktu eksekusi: 4474 ms

Gambar 8 Waktu Eksekusi Boyer-Moore

Algoritma *bruteforce* melakukan perbandingan pada tiap karakter teks selama pola belum ditemukan. Algoritma Knuth-Morris-Pratt dapat melewati beberapa perbandingan yang sebenarnya tidak diperlukan dengan cara membuat tabel fungsi pinggir pada saat prakomputasi. Sedangkan, algoritma Boyer-Moore bahkan dapat menghindari lebih

banyak perbandingan lagi, karena perbandingan dimulai dari kanan ke kiri, bukan kiri ke kanan.

Tetapi apabila variasi dari karakter kecil, algoritma Boyer-Moore tidak dapat bekerja secara efisien. Hal ini dikarenakan jarak lompat bertambah seiring dengan panjang pola. Algoritma Boyer-Moore dapat bekerja dengan baik ketika teks dan pola yang diproses menyerupai bahasa natural manusia, contohnya bahasa Inggris. Sedangkan, algoritma Knuth-Morris-Pratt lebih sesuai apabila digunakan untuk mencari pola dalam teks yang variasi karakternya lebih kecil.

## V. KESIMPULAN

Pencarian informasi pada sebuah basis data dapat menggunakan algoritma pencocokan pola seperti *bruteforce*, Knuth-Morris-Pratt, dan Boyer-Moore. Awalnya, QR *code* dipindai dengan aplikasi QR *code scanner*, QR *code scanner online*, ataupun program yang dibangun sendiri dan didukung oleh *library* tertentu. Kemudian, sistem akan melakukan pencocokan pola untuk mencari indeks data yang sesuai pada basis data media sosial tersebut. Terakhir, sistem akan memanggil data pada indeks tersebut dan menampilkan ke layar pengguna yang melakukan request lewat pemindaian QR *code* yang dicari.

Dengan menggunakan algoritma pencocokan pola yang lebih tepat dibanding *bruteforce*, waktu eksekusi dan sumber daya yang harus disediakan oleh sistem untuk melakukan pencarian pola pada QR *code* dapat berlangsung lebih mangkus. Hal ini menyebabkan pencarian informasi dalam basis data lebih cepat, juga mempersingkat waktu respon yang diberikan oleh sistem yang menggunakan basis data tersebut.

## UCAPAN TERIMA KASIH

Pertama-tama, penulis memanjatkan puji syukur kepada Tuhan yang Maha Esa atas berkat dan rahmat-Nya sehingga penulis dapat menyelesaikan makalah ini tepat waktu. Terima kasih juga penulis sampaikan kepada orang tua dan keluarga, yang selalu mendukung, mendoakan, dan

memberi semangat kepada penulis dalam menjalani kuliah. Selain itu, penulis juga menyampaikan ucapan terima kasih kepada pengampu mata kuliah IF2211 Strategi Algoritma yaitu Dr. Nur Ulfa Maulidevi S.T., M.Sc., Dr. Masayu Leylia Khodra M.T., dan Dr. Ir. Rinaldi Munir, M.T. atas bimbingan, dukungan, referensi, dan penyampaian ilmu yang telah membuat penulis mampu menulis makalah ini. Terakhir, terima kasih kepada rekan-rekan dan pihak lain yang membantu dalam proses pembuatan makalah ini.

## REFERENSI

- [1] <https://web.archive.org/web/20130129064920/http://www.qrcode.com/en/qrfeature.html>. Diakses pada 16 Mei 2017 pukul 22.25.
- [2] <https://zxingnet.codeplex.com/>. Diakses pada 17 Mei 2017 pada pukul 09.33.
- [3] Silberschatz, et.al. 2011. Database System Concepts 6<sup>th</sup> Edition.
- [4] Munir, Rinaldi. 2004. Diktat IF2211 Strategi Algoritmik: Algoritma Pencarian String (*String Matching*).
- [5] <https://www.quora.com/How-do-QR-code-generators-work>. Diakses pada 16 Mei 2017 pada pukul 22.30.

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 17 Mei 2017



Holy Lovenia - 13515113