Analisis Metode dan Penerapan Konsep *Digital Watermarking* pada QRIS (*Quick Response Code Indonesia Standard*)

David Petra Natanael
18217011
Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
Bandung, Indonesia
petragtr8@gmail.com

***Abstrak*—*QR Code* adalah salah satu kanal pembayaran berjenis *Shared Delivery Channel* yang mampu digunakan secara massal. Penggunaan *QR Code* dalam pembayaran semakin melonjak akibat perusahaan teknologi finansial. Bank Indonesia dengan 5 visi sistem pembayaran Indonesia di tahun 2025 pun meluncurkan standarisasi *QR Code* bernama QRIS. Sebuah sistem baru ini masih memiliki celah keamanan yang bisa dimanfaatkan oleh peretas. Untuk mencegah hal tersebut, akan dilakukan penerapan konsep *digital watermarking* untuk mengamankan sebuah QRIS supaya tidak dicurigai sebagai kode yang palsu. Metode *digital watermarking* yang digunakan pun akan dianalisis dalam makalah ini untuk melihat kecocokannya pada pengamanan QRIS.**

Kata Kunci*—*QR Code, QRIS, Digital Watermarking, Embedding & Extracting, OTP

I. Pendahuluan

Perkembangan teknologi yang pesat memungkinkan berbagai hal menjadi lebih praktis. Salah satu kegiatan yang terdampak adalah cara seseorang melakukan transaksi pembayaran. Perlahan demi perlahan sistem pembayaran tunai mulai terkisis dan kini dimungkinkan untuk melakukan transaksi secara non-tunai/*cashless*, artinya tanpa ada uang secara fisik seseorang dapat menyelesaikan transaksinya secara digital.

Indonesia pun memiliki sejarah dari segi penggunaan sistem pembayaran non-tunai. Metode pembayaran non-tunai terus berevolusi untuk menyempurnakan fungsinya sebagai alat pembayaran yang sah. Mulai dari awal perkembangannya yaitu pembayaran yang dilakukan lewat ATM hingga akhirnya muncul perkembangan mutakhir, dimana muncul teknologi pembayaran massal menggunakan *Quick Response Code (QR Code)*.

Sejalan dengan itu, sekitar 5 tahun belakangan ini pun tak terasa bahwa ada juga perkembangan perusahaan finansial berbasis teknologi *(fintech)* seperti GoPay, OVO, DANA, LinkAja, dll. Perusahaan teknologi finansial yang juga menyediakan layanan pembayaran non-tunai menggunakan uang elektronik pun memanfaatkan pembayaran dengan *QR Code*. Kecepatan dan kemudahan transaksi yang ditawarkan oleh pembayaran menggunakan *QR Code* ini membuat adanya lonjakan besar dalam transaksi non-tunai. Bank Indonesia mencatat bahwa sampai bulan September 2019 volume transaksi menggunakan uang elektronik mencapai 2,7 miliar transaksi dan nilai transaksi yang berhasil dicatat sebesar 95,7 triliun rupiah. Angka ini pasti akan terus bertambah kedepannya.

Melihat hal tersebut, Bank Indonesia dan Asosiasi Sistem Pembayaran Indonesia (ASPI) meluncurkan standarisasi untuk pembayaran menggunakan *QR Code* yakni dengan meluncurkan QRIS *(Quick Response Code Indonesia Standard)* yaitu sebuah *QR Code* yang sama untuk seluruh pembayaran. Peluncuran QRIS ditujukan agar pembayaran non-tunai bisa lebih mudah dilakukan karena setiap penyedia layanan tidak perlu menggunakan *QR Code* yang berbeda serta memudahkan regulator untuk mengawasi dari satu pintu saja.

Namun peluncuran QRIS ini tidak serta merta menyelesaikan seluruh masalah yang ada. QRIS masih perlu diwaspadai keamanannya. Sebab pada dasarnya QRIS juga merupakan *QR Code* statis yang biasa digunakan selama ini. Menurut Spesialis Keamanan Teknologi Vaksincom dan salah satu Peneliti Keamanan Siber Communication Information System Security Research Center (CISSReC) [7], terdapat bentuk kejahatan yang mungkin terjadi seperti pemalsuan dan modifikasi kode QR, dimana disematkan URL lain atau bahkan *malware* sehingga uang yang seharusnya ditransfer ke pihak uang elektronik malah masuk ke rekening peretas ataupun juga mengarahkan pengguna ke web yang berbahaya.

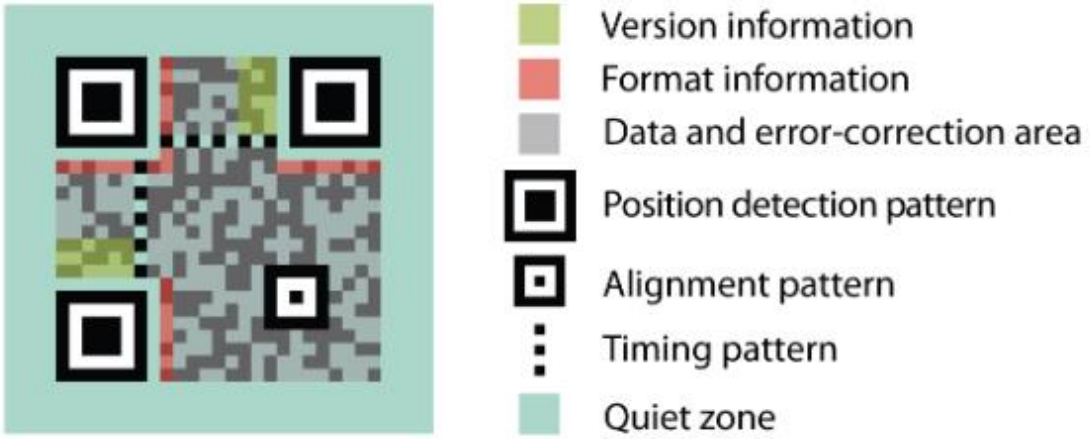
Berdasarkan hal tersebut, pada makalah ini kemudian akan dibahas bagaimana cara yang dapat dilakukan untuk lebih mengamankan sebuah *QR Code* terutama QRIS dengan menggunakan sebuah teknik bernama *watermarking*. Akan dilakukan analisis metode *watermarking* beserta penerapannya yang paling tepat dan cocok digunakan untuk mengamankan QRIS.

II. Dasar Teori

A. *QR Code* dan QRIS

i. *QR Code*

Berdasarkan pengembang awalnya [1] [6], *QR Code* adalah sebuah bentuk kode dua dimensi hasil evolusi dari kode batang satu dimensi yang biasa dibilang *barcode*. *QR Code* berbentuk matriks yang dapat dibaca secara horizontal maupun vertikal sehingga memungkinkan penampungan data yang lebih banyak. Dikembangkan oleh perusahaan Denso Wave dari Jepang mulai tahun 1994, asal nama *QR Code* yaitu *Quick Response* memang ditujukan supaya *QR Code* mudah dan cepat dibaca/dipindai. Struktur *QR Code* umumnya dibangun dari pola kotak berwarna hitam dan detailnya dapat dilihat pada Gambar 1.



Gambar 1 Struktur *QR Code* [6]

Setiap elemen berwarna hitam dan putih disebut *modules*. Semua elemen tersebut terletak secara horizontal dan vertikal. Kemudian setiap elemen hitam dan putih ini merepresentasikan 0 atau 1, sehingga tiap *module* ini dapat dibaca oleh komputer. *Module* pada *QR Code* bisa melakukan beberapa fungsi. Ada yang merepresentasikan data itu sendiri (ditandai dengan *data and error-correction area*), sisanya fungsi di grup ke beberapa pola yang bisa meningkatkan skala dan kinerja pemindaian lewat teknik *symbol alignment* (ditandai dengan *alignment pattern*), *error correction* (ditandai dengan *format information*), dan *distorsion compensation* (*alignment pattern* juga). Untuk area *version information, QR Code* dapat menghasilkan 40 simbol versi yang berbeda mulai dari ukuran 21x21 (versi 1) hingga 177x177 (versi 40).

Kemudian area *timing pattern* memungkinkan pemindai mengetahui besar matriks *QR Code*. Selanjutnya bagian *quiet zone* adalah area wajib kosong tanpa data untuk memisahkan area sekitar *QR Code* supaya tidak dideteksi oleh pemindai. Dibutuhkan beberapa waktu untuk memindai seluruh matriks 2 dimensi dalam mencari simbol/*module* yang tepat untuk mendefinisikan orientasi, posisi, dan besaran data. Maka dari itu *QR Code* didesain dengan sebuah pola pendeteksi posisi yang ditempatkan di 3 ujung kode (ditandai dengan *position detection pattern*). Rasio yang dimiliki oleh *position detection pattern* adalah 1:1:3:1:1 yang artinya kode dapat dipindai dari berbagai arah dan dalam sudut 360 derajat.

Dengan fitur tersebut, akan terbentuk relasi antara pola pendeteksi posisi yang memungkinkan akses cepat ke orientasi, posisi, dan besaran data. Denso Wave [6] pun mengklaim bahwa pemindaian *QR Code* dengan struktur seperti itu dapat mempercepat pemindaian sebesar 20 kali lebih cepat dari pembacaan kode matriks biasanya.

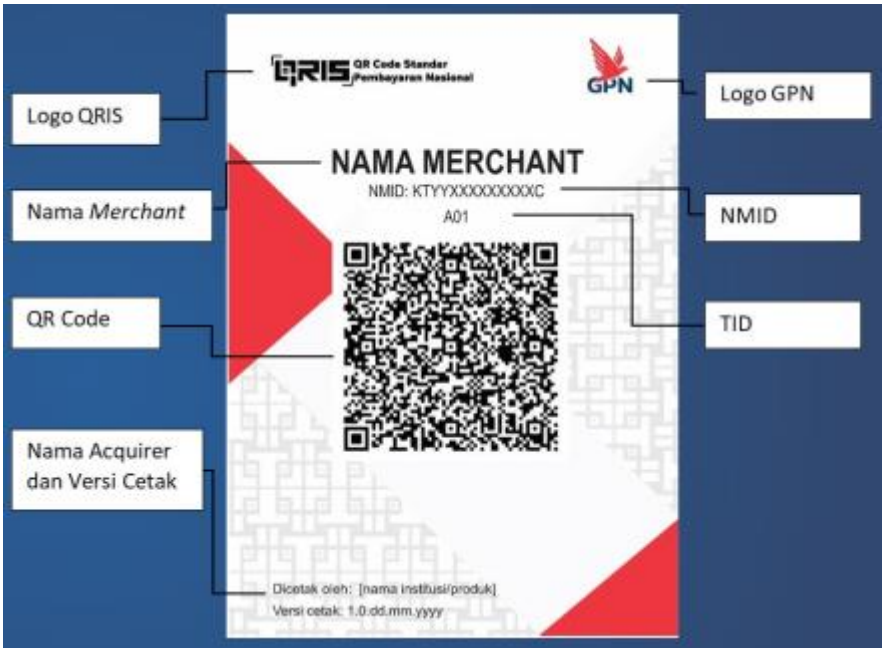
ii. *QR Indonesia Standard* (QRIS)

QRIS dikembangkan oleh Bank Indonesia dan Asosiasi Sistem Pembayaran Indonesia (ASPI) menggunakan standar internasional EMV Co. (lembaga penyusun standar internasional *QR Code* pembayaran). Diluncurkan sejak 17 Agustus 2019 kemudian baru wajib digunakan oleh seluruh penyedia layanan pembayaran non-tunai terhitung 1 Januari 2020. Sebelumnya telah dilakukan pengembangan dan uji coba layanan QRIS sejak September 2018. QRIS sendiri menjadi standar yang menggantikan banyaknya *QR Code* yang dibuat oleh Penyelenggara Jasa Sistem Pembayaran (PJSP) non-tunai di Indonesia.

Berdasarkan mekanisme dan peraturan yang telah dibuat oleh BI dan ASPI [8], QRIS memiliki 2 jenis model pembayaran yaitu *Merchant Presented Mode* (MPM) dan *Customer Presented Mode* (CPM). Model MPM menempatkan QRIS langsung di *merchant* terkait untuk selanjutnya dipindai oleh pengguna, sementara model CPM adalah dimana penggunanya yang menunjukkan QRIS untuk dapat dipindai oleh mesin pemindai *QR Code*.

Model MPM membuat transaksi dilakukan secara *push payment*, artinya transaksi dipicu oleh transfer yang dilakukan dari akun pengguna PJSP. MPM dibagi jadi 2 jenis yaitu *static* dan *dynamic*. Jenis *static* adalah QRIS yang berupa stiker/*print-out* dan ditempelkan pada *merchant* terkait (*generate* satu kali). Biasanya

digunakan oleh UMKM dan memiliki ID *merchant* yang tetap. Struktur QRIS *static* dapat dilihat pada Gambar 2. Sementara jenis *dynamic* adalah QRIS yang dibuat secara *real-time* pada saat transaksi menggunakan mesin EDC sehingga hasil kodenya berbeda-beda tiap saat. Jenis *dynamic* digunakan oleh usaha menengah dan besar.



Gambar 2 Struktur QRIS *Static* [8]

Model CPM membuat transaksi dilakukan secara *pull payment*, artinya pihak *merchant* menggunakan jasa *acquirer* untuk menagihkan pembayaran ke akun pengguna PJSP. Model ini membutuhkan standar untuk kode QR yang ditampilkan oleh pengguna, pemindai, hingga aplikasi POS sebagai alat penagih pembayaran. Model CPM juga sesuai untuk usaha menengah dan besar.

B. *Digital Watermarking*

Watermarking sendiri adalah salah satu teknik menyembunyikan informasi dalam suatu medium untuk melindungi informasi di dalamnya dengan sebuah tanda khusus. Tujuan *watermarking* adalah menjaga paten, digunakan untuk autentikasi, serta menjaga integritas suatu data [5].

Kini *watermarking* sudah dapat diaplikasikan ke media digital dan bisa diaplikasikan pada teks, gambar, audio, hingga video. Spesifik masuk ke dalam tipe *digital image watermarking*, *watermark* yang dapat diterapkan pada medium sebuah gambar adalah sinyal kecil seperti *noise* yang dibenamkan ke gambar yang ingin dilindungi. Konten dari *watermark* bisa berupa *string of bits* yang merepresentasikan pemiliknya.

Proses pemberian *watermark* juga diharuskan hanya menggunakan *noise* atau distorsi yang relatif kecil sehingga mencegah hilang/sulitnya dibaca data awal.

Proses *watermarking* dibagi menjadi 2 tahap yaitu tahap enkripsi dan dekripsi. Dalam tahap enkripsi, *watermark* yang ingin dibenamkan (W) akan digabungkan dengan sebuah algoritma khusus. Gabungan kedua hal tersebut lalu barulah dibenamkan dalam gambar (I) sehingga keluarannya adalah *watermarked image* (I_w). Sementara untuk tahap dekripsi, *watermarked image* (I_w) akan diterima oleh algoritma dekripsi. Dalam proses ini, kadang ada algoritma yang membutuhkan gambar asli (I) dan ada yang tidak. Hasil keluaran proses dekripsi adalah *watermark* (W) yang tadi telah ditambahkan.

Terdapat 2 teknik yang diperlukan untuk membuat sebuah *watermark* [5] yaitu metode domain spasial (*spatial domain method*) dan metode domain transformasi (*transform domain method*). Pada teknik spasial, integrasi *watermark* dilakukan dengan melakukan modulasi intensitas *pixel* dari gambar yang dipilih. Caranya dengan memodifikasi/menempatkan perubahan intensitas di *Least Significant Bit* (LSB) dari gambar. Sementara teknik transformasi, sesuai namanya menggunakan sebuah transformasi DCT (*Discrete Cosine Transform*). Transformasi dilakukan dengan mendistribusikan *watermark* ke seluruh bagian gambar secara irregular, sehingga akan sulit dideteksi. Ini membuat *watermark* dengan teknik transformasi lebih popular dan lebih aman karena gangguan sinyal sulit dideteksi pada gambar akibat distribusi tadi.

III. Metode dan Persiapan *Digital Watermarking*

Sebelum menerapkan konsep *watermarking* pada QRIS, diperlukan pemilihan dan penjelasan mengenai metode yang tepat dan cocok. Metode ini akan menggabungkan beberapa konsep yang harus dipersiapkan terlebih dahulu untuk menjamin keamanan pada QRIS.

A. Jenis *Watermarking*

Pertama yang dipilih adalah jenis *watermarking*. Akan digunakan jenis *transform domain watermarking* yaitu DCT *Watermarking*. Pertimbangan penggunaan konsep tersebut adalah [4] sudah banyak pemakaiannya pada pemrosesan gambar dan sinyal dengan berbagai keunggulannya seperti bisa melakukan kompresi yang tinggi, *error rate* yang kecil, dan memiliki integritas informasi yang tinggi. [4] Transformasi DCT untuk *image encoding* (DCT 2 dimensi) dapat didefinisikan sebagai (1):

$$f(x,y) = C(u)C(v)\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)\cos\left[\frac{(2x+1)u\pi}{2N}\right] cos\left[\frac{(2y+1)v\pi}{2N}\right] \qquad (1)$$

Sementara untuk *decoding/inverse* DCT dapat didefinisikan sebagai (2):

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] cos\left[\frac{(2y+1)v\pi}{2N}\right] \qquad (2)$$

Dimana pada $C(\omega) = \begin{cases} 1/\sqrt{2}, & \omega = 0 \\ 1, & \omega = 1,2,3,\dots,N-1 \end{cases}$, N adalah total *pixel* dari blok *pixel* horizontal dan vertikal, yaitu N = 8. Kemudian F(u,v) adalah koefisien transformasi *cos*, dimana u dan v adalah variabel frekuensi (u,v = 1,2,3,…,N-1). Jika f(x,y) adalah urutan domain waktu M dan N, maka x dan y dapat bernilai 1,2,3,…,N-1.

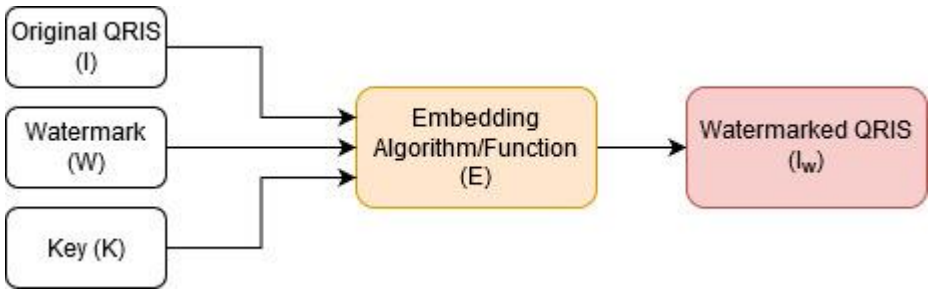
Fungsi (1) ini akan digunakan dalam proses *encoding* untuk menghasilkan QRIS yang telah dibenamkan *watermark*. Sementara untuk fungsi (2) akan digunakan dalam proses *decoding* untuk memverifikasi hasil pindaian QRIS apakah asli atau palsu.

B. *Embedding Watermark*

Kedua adalah konsep untuk membenamkan *watermark* yakni dengan memodelkan proses tersebut sebagai fungsi E (*embedding function*) [3] berikut ini (3):

$$I_w = E(I,W) = E(I,F(I,W,K)) \qquad (3)$$

Dimana I_w adalah QRIS yang sudah diberi *watermark*, I adalah gambar asli QRIS, dan W adalah *watermark* yang ingin ditambahkan. W adalah hasil dari fungsi yang mengkalkulasikan penggabungan I yaitu gambar asli QRIS, W yaitu sinyal *watermark*, dan K yaitu *key*.



Gambar 3 Proses *Embedding Watermark* [3]

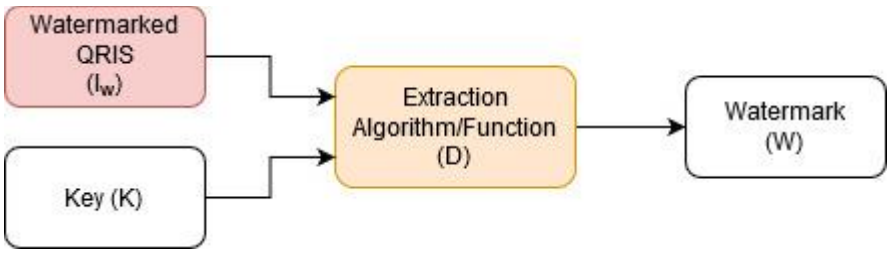
Skema model tersebut bisa dilihat pada Gambar 3, dimana akan ada *watermark* digital yang dibenamkan ke kode QRIS dengan fungsi E lalu ditambahkan dengan *key*. Hasilnya adalah QRIS yang berhasil diberikan *watermark*. Fungsi E ini adalah referensi dari fungsi (1) pada bagian sebelumnya yaitu untuk melakukan *encode watermark*.

C. *Extracting Watermark*

Lalu yang ketiga adalah konsep untuk melakukan ekstraksi *watermark* ketika telah diterima oleh *user*. Konsep ini juga dapat dimodelkan lewat fungsi D (*decode/extraction function*) [3] berikut ini (4):

$$W = D(I_w, K) \qquad (4)$$

Dimana W adalah *watermark* yang sudah ditambahkan sebelumnya, I_w adalah *watermarked* QRIS, dan K adalah *key*.



Gambar 4 Proses *Extraction/Decode Watermark* [3]

Konsep ekstraksi *watermark* untuk model fungsi D tersebut berasal dari jenis ekstraksi *blind watermark* (Gambar 4), artinya tidak diperlukan data asli untuk melakukan ekstraksi *watermark*. Sehingga ketika masuk ke algoritma, maka akan langsung keluar hasil informasi *watermark* tadi. Fungsi D adalah referensi dari fungsi (2) pada bagian sebelumnya yaitu untuk melakukan *decode watermark*.

D. Isi *Watermark*

Untuk menandakan bahwa QRIS yang dipindai oleh pengguna saat melakukan transaksi di PJSP nantinya adalah milik Bank Indonesia, maka Bank Indonesia perlu menentukan juga isi *watermark* yang diinginkan. *Watermark* bisa berbentuk teks, gambar, audio, ataupun video.

Pada penerapannya, *watermark* yang mungkin lebih tepat untuk dibenamkan pada QRIS adalah gambar karena gambar bisa lebih merepresentasikan kepemilikan dan

keaslian/keabsahan QRIS. Bank Indonesia dapat membenamkan sebuah gambar khusus pada QRIS layaknya mencetak uang yang juga punya penanda khusus. Maka dari itu akan diasumsikan bahwa Bank Indonesia menggunakan gambar khusus yang bisa dijadikan *watermark* pada QRIS.

E. Penggunaan *Key* Berbasis OTP

Penambahan *key* ditujukan untuk meningkatkan keamanan agar mencegah pihak lain/peretas yang ingin memanipulasi kode QRIS maupun *watermark* itu sendiri [2]. Sebenarnya *key* bisa dibuat dengan berbagai metode karena yang menjadi poin penting disini adalah tujuan penggunaan *key* bisa menambah autentikasi pada proses pemindaian kode QR. Sebuah *key* juga tidak harus dibuat berkorelasi dengan kode QR. Akan lebih baik jika *key* ini di-*generate* secara acak.

Salah satu konsep *key* yang akan digunakan adalah OTP [2]. OTP (*One Time Password*) seperti namanya, adalah sebuah jenis *password* sekali pakai yang unik dan biasa digunakan dalam proses *login*. Kegunaan utamanya ialah untuk memberikan *password* sekali pakai yang akan terus berganti setiap *password* itu telah dipakai atau melewati batas pemakaiannya. Hal ini memberikan keamanan ekstra pada suatu sistem dan akan sulit dilacak juga oleh peretas.

Konsep OTP yang digunakan akan mengacu pada jenis *time-based* OTP, yaitu kode OTP yang akan terus berganti untuk setiap interval waktu tertentu apabila tidak digunakan. Konsep ini dipertimbangkan karena sudah banyak diterapkan dalam metode *login* maupun pembayaran pada bank. Selain itu dari segi kesediaan alat dan infrastruktur, konsep OTP bisa diterapkan secara langsung karena kode OTP bisa dikirimkan ke nomor ponsel pengguna yang sudah terdaftar pada setiap PJSP. Nantinya pembuatan *watermarked* QRIS akan menggunakan OTP sebagai *key* yang harus disamakan ketika proses transaksi berlangsung.

IV. Penerapan Pada QRIS

QRIS yang diedarkan oleh Bank Indonesia memiliki 2 model pembayaran yaitu MPM dan CPM sesuai dengan yang telah dijelaskan pada bagian 2. Di dalam model pembayaran MPM atau yang berbasis dengan *merchant*, terdapat QRIS statis dan dinamis. Maka dari itu total terdapat 3 cara QRIS diimplementasikan pada transaksi pembayaran. Dari ketiga cara ini, perlu dilakukan juga penerapan yang berbeda-beda karena satu cara pembayaran tidak sama dengan cara lainnya.

A. Dynamic QRIS pada MPM

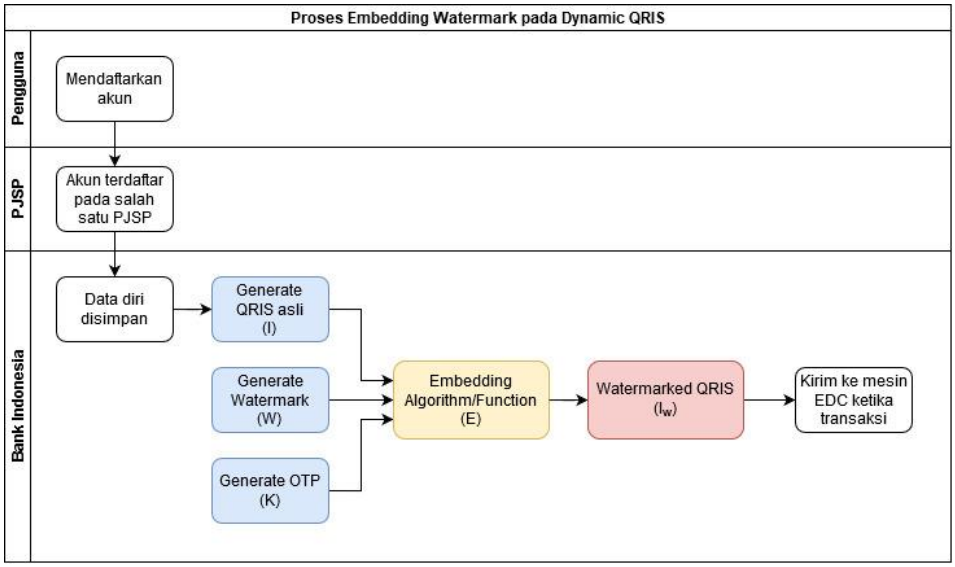
Proses penerapan *watermark* pada QRIS yang dihasilkan secara dinamis oleh mesin EDC bisa dilakukan juga dengan metode pada bagian 3. Mula-mula adalah fase *embedding watermark*:

1. Asumsikan pengguna aplikasi PJSP telah memiliki akun. Setiap pengguna tentunya telah mendaftarkan data dirinya seperti nama, *email*, nomor telepon, ID akun, dll.
2. Setiap data diri pengguna akan disimpan oleh PJSP dan Bank Indonesia dikhususkan berhak untuk menggunakan data diri ini dalam pembuatan *watermark*.
3. Proses *embedding watermark* dilakukan oleh pihak Bank Indonesia dimana akan ada 2 jenis QRIS yang dihasilkan, 1 adalah QRIS asli yang dibuat berdasarkan data diri, 1 lagi adalah QRIS hasil pemberian *watermark*.
4. QRIS yang dibuat berdasarkan data diri akan menjadi ***cover image*** untuk input masuk saat proses *encoding*, dalam hal ini QRIS ini adalah Original QRIS (I) pada Gambar 3). QRIS asli akan disimpan oleh pihak Bank Indonesia.
5. Nantinya pada setiap transaksi yang terjadi, akan dihasilkan QRIS dari mesin EDC. Hasil QRIS ini adalah QRIS yang sudah diberikan *watermark* (I_w), dimana terjadi proses seperti Gambar 3 yaitu adanya *encoding* antara Original QRIS (I), *watermark* gambar dari Bank Indonesia (W), serta *key* (K) yaitu OTP.
6. OTP di awal *encoding* di-*generate* oleh Bank Indonesia di awal pembentukan *watermark* agar nantinya akan dicek kesamaannya ketika pengguna melakukan transaksi.

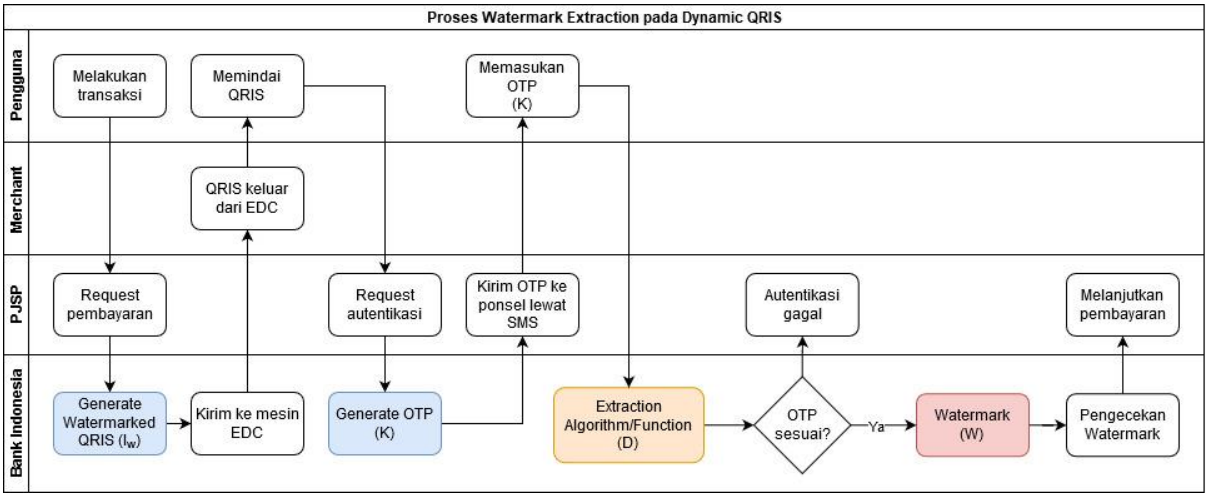
Selanjutnya adalah proses *watermark extraction*:

1. Ketika transaksi dilakukan dengan uang elektronik, mesin EDC akan mengeluarkan QRIS untuk selanjutnya dipindai. QRIS ini sudah dalam bentuk *watermarked*.
2. Pengguna akan menggunakan salah satu aplikasi PJSP dan melakukan pemindaian pada QRIS.
3. Saat pemindaian dilakukan, akan terjadi proses *watermark extraction*. Hal ini dilakukan untuk melihat apakah QRIS yang di-*generate* adalah QRIS asli atau bukan.
4. Pengecekan keaslian QRIS ditandai dengan dikirimnya kode OTP ke ponsel pengguna lewat SMS.
5. Apabila pengguna memasukan kode OTP yang tepat saat proses pemindaian, ini artinya fungsi *extraction* bisa berjalan dan akan keluar *watermark* yang dibenamkan sebelumnya.

6. *Watermark* yang keluar menjadi penanda bahwa kode QRIS asli atau bukan. Bila *watermark* yang keluar masih utuh atau dalam keadaan yang baik (pengecekan diasumsikan dilakukan otomatis oleh sistem), maka transaksi pembayaran bisa dilakukan. Apabila *watermark* rusak di tengah jalan, ini artinya transaksi akan diberhentikan akibat *watermark* yang rusak.



Gambar 5 Proses *Embedding Watermark* pada *Dynamic* QRIS



Gambar 6 Proses *Watermark Extraction* pada *Dynamic* QRIS

Dikarenakan kode QRIS yang dihasilkan dari setiap transaksi berbeda-beda, jenis QRIS dinamis ini bisa dibilang memiliki tingkat keamanan yang lebih tinggi karena peretas akan sulit untuk melakukan modifikasi pada QRIS yang berbeda-beda setiap transaksinya.

B. Static QRIS pada MPM

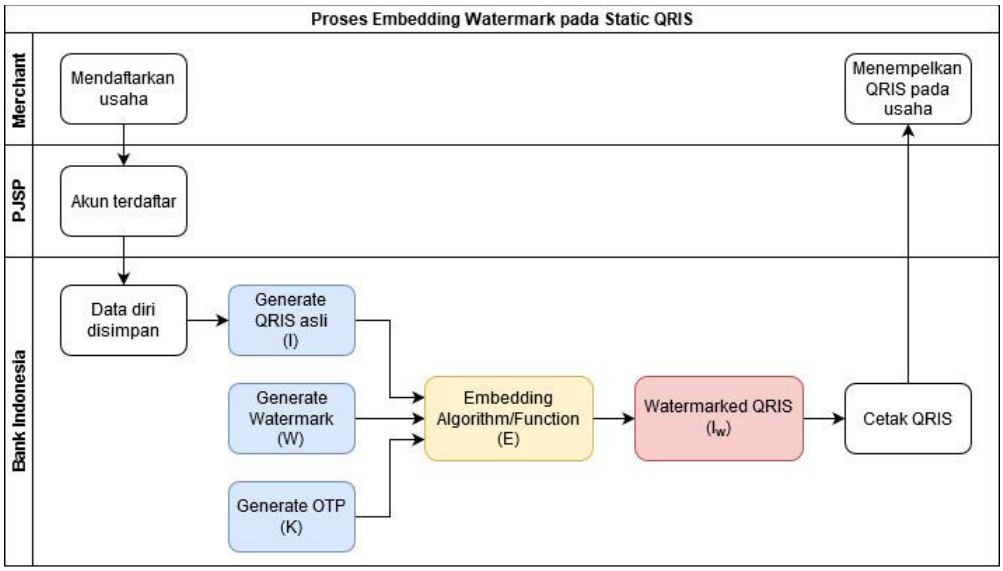
Terdapat perbedaan pada QRIS statis, dimana pada cara ini, setiap *merchant* sudah memiliki ID dan QRIS-nya masing-masing. Hal ini membuat tidak adanya QRIS yang unik pada setiap transaksi. Maka dari itu akan diterapkan konsep yang sedikit berbeda dengan penerapan *watermark* pada *dynamic* QRIS. Pertama adalah fase *embedding watermark*.

1. *Merchant* mendaftarkan usahanya langsung pada PJSP. Bank Indonesia akan menggunakan data tersebut untuk membuat QRIS statis yang siap ditempel di tempat *merchant.*
2. Data akan menjadi QRIS statis yang digabungkan dengan *watermark* serta *key* dari Bank Indonesia. Kemudian baru dibentuk *watermarked* QRIS.
3. Selanjutnya QRIS hasil *watermark* akan dicetak dan diberikan kepada *merchant* terkait. QRIS ini tidak berbeda-beda untuk setiap transaksi.

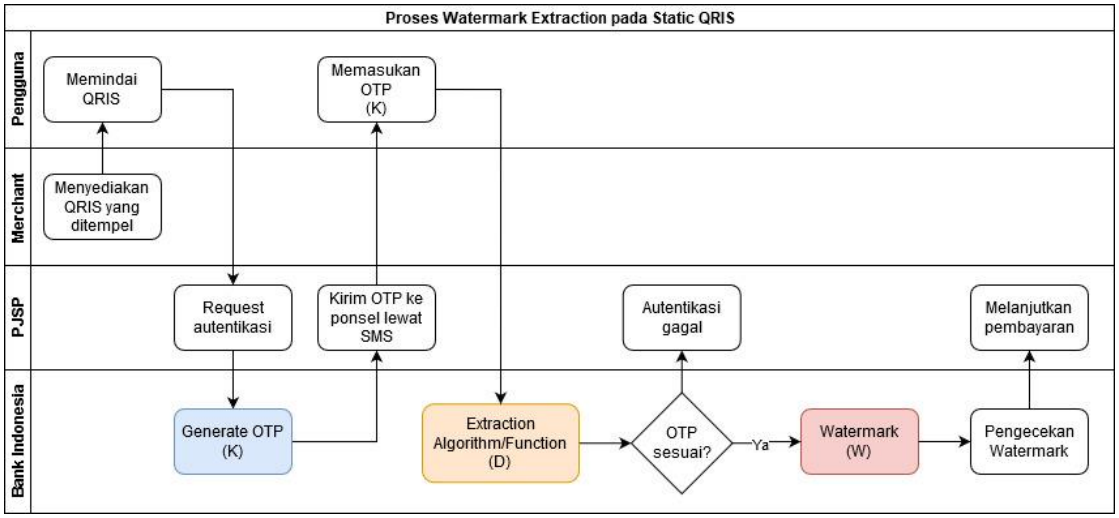
Sementara untuk fase *watermark extraction*:

1. Transaksi dilakukan saat pengguna ingin melakukan pemindaian pada QRIS *merchant*.
2. Pengguna yang memindai QRIS selanjutnya akan dikirimkan kode OTP untuk proses ekstraksi *watermark*.
3. Apabila OTP sesuai maka *watermark* berhasil keluar dan akan dilakukan pengecekan lebih lanjut secara otomatis.

Kode QRIS yang ada pada *merchant* adalah statis, akan tetapi tingkat keamanannya tidak setinggi *dynamic* QRIS yang selalu berubah-ubah setiap transaksinya.



Gambar 7 Proses *Embedding Watermark* pada *Static* QRIS



Gambar 8 Proses *Watermark Extraction* pada *Static* QRIS

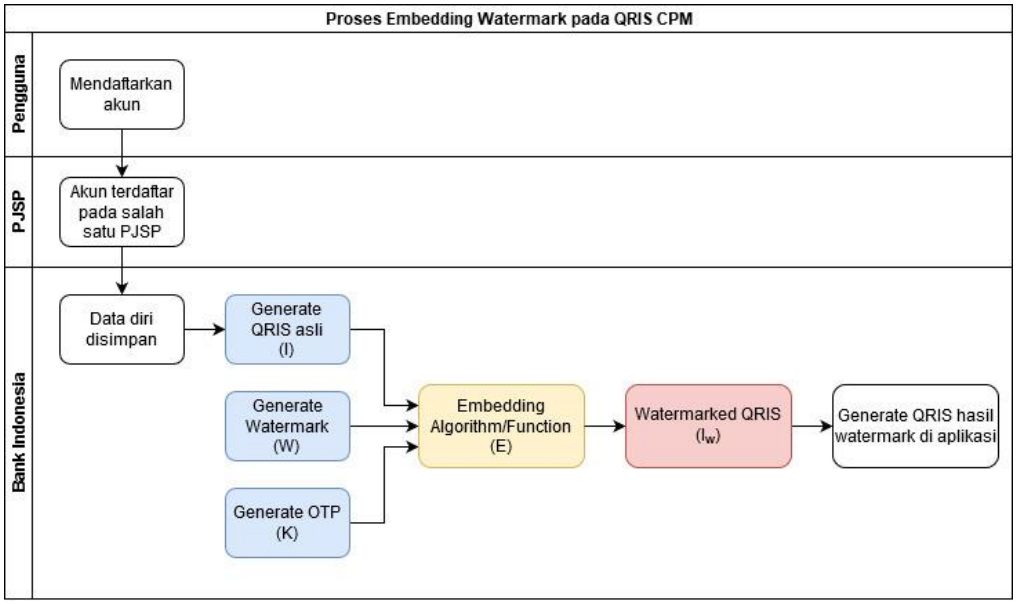
C. QRIS pada CPM

QRIS tipe CPM adalah QRIS yang ditunjukkan oleh pengguna saat melakukan transaksi pada *merchant*. QRIS akan dipindai oleh sistem POS yang ada pada *merchant*. Ketika melakukan transaksi, QRIS yang ditunjukkan pengguna dapat di-*generate* secara berubah-ubah untuk setiap transaksinya dan QRIS ini adalah QRIS yang sudah diberikan *watermark*. Proses *embedding watermark*-nya adalah sebagai berikut.

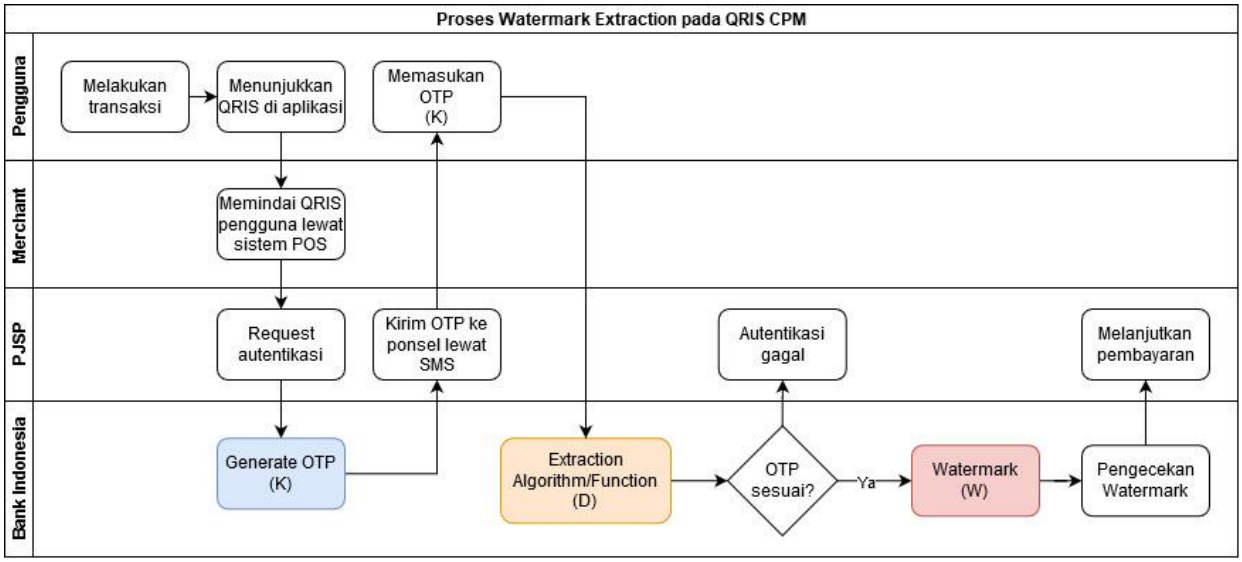
1. Pengguna mendaftarkan akun pada PJSP. Data diri dipakai oleh Bank Indonesia untuk membuat *cover image* yaitu QRIS pengguna.
2. Sama seperti sebelumnya, akan dilakukan proses *encoding* dengan menggabungkan QRIS asli dari data diri, *watermark*, dan *key* OTP.
3. Setelah proses tersebut selesai, terbentuk *watermarked* QRIS yang akan di-*generate* setiap transaksi ataupun ketika pengguna memilih metode pembayaran dengan menunjukkan kodenya.

Sementara untuk proses *watermark extraction*:

1. Pengguna membuka salah satu aplikasi PJSP dan menunjukkan QRIS yang di-*generate* lewat metode pembayaran yang dipilih.
2. *Merchant* akan melakukan pemindaian QRIS menggunakan sistem POS.
3. Akan dikirimkan kode OTP ke ponsel pengguna lewat SMS.
4. Pengguna memasukan kode tersebut, apabila sesuai maka transaksi bisa dilanjutkan sesuai dengan pengecekan *watermark* secara otomatis.



Gambar 9 Proses *Embedding Watermark* pada QRIS Model CPM



Gambar 10 Proses *Watermark Extraction* pada QRIS Model CPM

QRIS yang ada pada model CPM juga dibuat berbeda-beda untuk setiap transaksi sehingga tingkat keamanannya juga lebih tinggi dari QRIS statis. Ini membuat peretas sulit mendeteksi QRIS tersebut, terlebih QRIS ini hanya ditampilkan oleh pengguna ketika ingin melakukan transaksi.

V. Kesimpulan

Tren metode pembayaran non-tunai menggunakan *QR Code* sudah semakin meningkat lewat banyaknya penyedia jasa sistem pembayaran (PJSP). Bank Indonesia pun mengeluarkan QRIS, yaitu standar kode yang akan menggantikan pembayaran dengan *QR Code* biasa. Namun, QRIS yang baru terbit ini masih perlu ditinjau segi keamanannya agar kode QRIS tidak mudah dipalsukan oleh peretas.

Konsep *digital watermarking* dapat menjadi alternatif untuk mengamankan kode QRIS. *Digital watermarking* yang diterapkan akan spesifik menggunakan metode DCT yaitu untuk membuat *watermark* pada gambar dua dimensi. DCT akan membutuhkan input berupa QRIS asli sebagai *cover image*, *watermark* khusus untuk menandakan kepemilikan, serta sebuah *key* dalam bentuk OTP. Hasilnya adalah gambar kode QRIS yang sudah ditambahkan *watermark*.

Untuk menjamin keaslian QRIS maka ketika transaksi dilakukan, pengguna akan memindai QRIS yang sudah diberikan *watermark*. Kemudian pengguna akan mendapatkan kode OTP yang berfungsi untuk melakukan *decode* pada QRIS. OTP menjadi salah satu autentikasi agar QRIS yang dipindai benar-benar asli atau tidak. Apabila kode OTP tepat, maka *watermark* bisa diekstrak untuk kemudian dicek secara otomatis keaslian dan kepemilikan *watermark* tersebut.

Dalam pemakaiannya, watermark pada kode QRIS akan disesuaikan dengan 3 metode pembayarannya yaitu QRIS statis dan dinamis pada metode MPM serta QRIS pada metode CPM. Masing-masing metode membutuhkan penanganan yang berbeda-beda.

Referensi

[1] Mishra and M. Mathuria, "Multilevel security feature for online transaction using QR code & digital watermarking," *2017 International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, 2017, pp. 48-51.

[2] J. Thomas and R. H. Goudar, "Multilevel Authentication using QR code based watermarking with mobile OTP and Hadamard transformation," *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Bangalore, 2018, pp. 2421-2425.

[3] S. Vongpradhip and S. Rungraungsilp, "QR code using invisible watermarking in frequency domain," *2011 Ninth International Conference on ICT and Knowledge Engineering*, Bangkok, 2012, pp. 47-52.

[4] L. Li, R. Wang, and C. Chang, "A Digital Watermark Algorithm for QR Code," *2011 International Journal of Intelligent Information Processing*, Vol. 2, Num. 2.

[5] J. Abraham, "Digital Image Watermarking: An Overview," *2011 Modern Trends in Electronic Communication & Signal Processing*, Kerala, 2011.

[6] Denso ADC, "QR Code Essentials," *Denso ADC QR Code White Paper r1f - NACS*, 2011. [Online]. Available: http://www.nacs.org/LinkClick.aspx?fileticket=D1FpVAvvJuo=&tabid=1426&mid=4802.

[7] F. A. Burhan, "Standardisasi Kode QR Berlaku 2020, Ahli IT: Keamanan Jadi Perhatian," *Berita Katadata.co.id*, 08-Jan-2020. [Online]. Available: https://katadata.co.id/berita/2020/01/08/standardisasi-kode-qr-berlaku-2020-ahli-it-keamanan-jadi-perhatian.

[8] Bank Indonesia, "QR Code Indonesian Standard (QRIS)," *Edukasi Sistem Pembayaran*, 13-Dec-2019. [Online]. Available: https://www.bi.go.id/id/sistem-pembayaran/edukasi/Pages/QR-Code-Indonesian-Standard.aspx.