Terakreditasi SINTA Peringkat 2
Surat Keputusan Dirjen Penguatan Riset dan Pengembangan Ristek Dikti No. 10/E/KPT/2019
masa berlaku mulai Vol. 1 No. 1 tahun 2017 s.d Vol. 5 No. 3 tahun 2021

Terbit online pada laman web jurnal: **http://jurnal.iaii.or.id**

JURNAL RESTI
(Rekayasa Sistem dan Teknologi Informasi)
Vol. 4 No. 2 (2020) 319 - 328 ISSN Media Elektronik: 2580-0760

Metode Pembayaran Elektronik yang Aman pada *Online Shopping* Berbasis Kriptografi Visual

Trihastuti Yuniati[1], Iqsyahiro Kresna A[2]
[1, 2] Program Studi S1 Teknik Informatika, Fakultas Informatika, Institut Teknologi Telkom Purwokerto
[1]trihastuti@ittelkom-pwt.ac.id, [2]hiro@ittelkom-pwt.ac.id

Abstract

Phishing and identity theft are common threats of online shopping. Phishing is an attempt to steal personal data from legitimate user. In this paper we propose a secure e-payment method using a credit card based on visual cryptography. This method adopts the existing 3D-Secure technology. Visual cryptography is applied in: user-merchant authentication, user-card provider authentication, user-card issuer authorization. It is applied to captcha image generated by merchant during registration using (2, 2) scheme with 2-subpixel expansion, to a text file containing credit card information sent by merchant to the card provider using (2, 2) scheme with pixel replacement, and to quick response code containing one-time-password that is used to authorize the payment transaction using (2, 2) scheme with 4-subpixel expansion. The test results show that out of 100 trials, all of them give 100% true positive. This indicates that the method is able to prevent phishing and identity theft, in sense of authentication, authorization, confidentiality, and integrity are gained. Phishing can be prevented because only legitimate participant who has an image share. Identity theft can be prevented because credit card details are not stored in the merchant's database. Authorization is more guaranteed because only authenticated user can authorize the payments.

Keywords: captcha, e-payment, one-time-password, quick response code, visual cryptography

Abstrak

Penipuan kartu kredit, *phishing* dan pencurian data pribadi merupakan ancaman pada jual-beli *online*. *Phishing* adalah upaya pencurian data-data pribadi dari pemilik sah. Pada paper ini diajukan metode pembayaran elektronik menggunakan kartu kredit pada jual-beli *online* berbasis kriptografi visual. Metode ini mengadopsi teknologi 3D-*Secure* yang berlaku saat ini. Kriptografi visual diterapkan di tiga fungsi, yaitu pengecekan keabsahan (autentikasi) pembeli-situs penjual, pembeli-penyedia layanan kartu kredit, dan pemberian ijin (otorisasi) pembeli kepada bank. Pertama, kriptografi visual diterapkan pada gambar *captcha* yang dibangkitkan oleh situs penjual ketika pembeli melakukan registrasi di situs penjual menggunakan skema (2, 2) dan ekspansi 2-subpiksel. Kedua, kriptografi visual diterapkan pada teks berisi informasi kartu kredit yang dikirimkan situs penjual kepada penyedia layanan kartu kredit menggunakan skema (2, 2) dengan penggantian piksel. Terakhir, kriptografi visual diterapkan pada *quick response code* yang menyimpan kode unik sekali pakai (*one-time-password*) untuk otorisasi pembayaran menggunakan skema (2, 2) dengan ekspansi 4-subpiksel. Sebuah *prototype* dibangun untuk simulasi dan pengujian keamanan. Hasil pengujian menunjukkan bahwa dengan 100 kali percobaan, keseluruhannya memberikan hasil 100% *true positif*. Artinya, metode yang diajukan dapat mencegah *phishing* dan pencurian data pribadi, serta autentikasi, otorisasi, kerahasiaan, dan integritas (keaslian) data lebih terjamin. *Phishing* dapat dicegah karena hanya pihak yang sah yang memiliki pecahan gambar hasil kriptografi visual. Pencurian data pribadi dapat dicegah karena data kartu kredit tidak disimpan di dalam basis data situs penjual. Otorisasi lebih terjamin karena hanya pengguna terautentikasi yang dapat melakukan transaksi dan mengotorisasi pembayaran.

Kata kunci: *captcha*, kriptografi visual, *one-time-password*, pembayaran elektronik, *quick response code*.

1. Pendahuluan

Semakin meluasnya penggunaan internet dalam kehidupan sehari-hari telah mengubah preferensi dan kebiasaan manusia, salah satunya di sistem jual-beli. Pada jaman dahulu orang-orang melakukan transaksi jual-beli secara langsung dengan tatap muka antara penjual dan pembeli. Saat ini orang-orang mulai beralih ke sistem jual-beli secara daring atau dikenal dengan *online shopping*. Sebagai implikasinya, sistem

Diterima Redaksi : 19-02-2020 | Selesai Revisi : 08-04-2020 | Diterbitkan Online : 20-04-2020

pembayaran elektronik menjadi hal yang penting. Salah satu metode pembayaran elektronik yang digunakan adalah pembayaran dengan kartu kredit. Namun demikian, berbagai insiden penipuan kartu kredit masih banyak terjadi dikarenakan masih adanya kelemahan pada sistem pembayaran elektronik.

Phishing dan pencurian data pribadi (*identity theft*) adalah ancaman yang sering terjadi pada sistem jual-beli *online*. *Phishing* adalah suatu upaya serangan dimana penyerang atau disebut *phisher* menyetting suatu situs web palsu yang terlihat serupa dengan situs web asli. Tujuan dari *phishing* adalah untuk memancing korban agar mengakses situs palsu tersebut dan memasukkan data-data kredensialnya, misal *username, password*, data kartu debit/kredit, dan disalahgunakan untuk meraih keuntungan pribadi. Sedangkan *identity theft* atau pencurian identitas juga merupakan upaya pencurian identitas. Upaya ini bisa dilakukan misalnya dengan membobol basis data server atau penyalahgunaan data oleh pihak internal.

Berbagai metode sudah diajukan untuk mengatasi permasalahan tersebut, namun metode-metode tersebut masih belum cukup efektif untuk mengatasi permasalahan yang terjadi, terbukti dengan masih banyaknya kasus penipuan kartu kredit atau yang dikenal sebagai *carding.* Oleh karena itu, pada paper ini diajukan suatu pendekatan baru untuk meningkatkan sistem pembayaran elektronik terutama yang menggunakan kartu kredit dengan menggunakan kriptografi visual. Studi kasus yang diambil dalam paper ini adalah pada sistem jual-beli secara daring melalui situs milik penjual atau *merchant.* Kriptografi visual akan diterapkan di tiga fase, yaitu pada saat registrasi *user* di situs *merchant*, saat verifikasi kartu kredit oleh *card provider*, dan selama proses otorisasi pembayaran antara *user*, *merchant*, dan *card issuer*. *Card provider* adalah penyedia jasa layanan kartu kredit, misalnya VISA, MasterCard, dll. *Card issuer* adalah bank yang mengeluarkan kartu kredit yang digunakan oleh pembeli atau *user*. Sedangkan bank yang terkait dengan merchant disebut sebagai *acquirer*.

1.1. Kriptografi Visual

Kriptografi visual adalah suatu teknik penyandian data yang dikembangkan oleh Naor dan Shamir pada tahun 1994. Pada teknik ini data disembunyikan di dalam suatu gambar, kemudian gambar tersebut dipecah menjadi 2 atau lebih potongan gambar yang disebut sebagai *share*, dan dengan suatu cara sedemikian sehingga ketika potongan-potongan *share* tersebut ditumpuk maka informasi yang tersembunyi di gambar tersebut akan dapat terlihat secara langsung oleh sistem visual manusia. *Share* adalah gambar piksel acak yang dibangkitkan dengan menggunakan algoritma kriptografi visual. Skema ini diklaim sangat aman dan mudah diimplementasikan. Karena proses dekripsi tidak memerlukan komputasi, skema ini sesuai digunakan pada sistem dengan spesifikasi *hardware* rendah. [1].

Model dasar dari skema penyandian data ini terdiri atas dua komponen, yaitu suatu cetakan teks yang disandikan atau disebut *ciphertext* dan suatu transparansi yang berfungsi sebagai kunci rahasia atau *secret key*. Kedua komponen ini pada dasarnya adalah potongan-potongan *share* yang terbentuk dari hasil pemecahan gambar asli. Pesan asli sebelum disembunyikan atau *plaintext* akan terlihat dengan meletakkan transparansi yang berisi *secret key* di atas *ciphertext*, meskipun masing-masing dari keduanya terlihat seolah seperti gambar *random noise* biasa [1].

Pada dasarnya terdapat beberapa jenis skema yang bisa digunakan pada algoritma kriptografi visual. Namun skema dasarnya adalah skema (2, 2). Skema (2, 2) artinya dari hasil operasi tersebut akan menghasilkan 2 potongan *share* hasil pecahan dari satu gambar, dimana *share* yang satu mengandung piksel random yang berisi *secret key* dan *share* yang lain menyimpan informasi rahasia atau *ciphertext.* Berdasarkan model dasar tersebut kemudian dikembangkan menjadi varian visual dengan k keluaran dari n bagian yang mengandung rahasia. Skemanya yaitu, dari suatu pesan dibangkitkan sebanyak n transparansi. Pesan asli akan terlihat jika sebanyak k (atau lebih) dari n transparansi yang dibangkitkan ditumpuk secara bersamaan, namun tidak akan terlihat sama sekali jika kurang dari k transparansi yang ditumpuk bersamaan. Skema ini disebut sebagai skema (k, n) [1].

Skema kriptografi visual yang paling sederhana mengasumsikan bahwa pesan tersusun atas sekumpulan piksel berwarna hitam dan putih, dimana masing-masing piksel dapat ditangani secara terpisah. Tiap piksel asli muncul di dalam n versi yang dimodifikasi, yang disebut *share*, sejumlah satu untuk masing-masing transparansi. *Share* merupakan kumpulan dari sejumlah m subpiksel hitam dan putih yang dicetak secara berdampingan satu sama lain sehingga sistem visual manusia dapat memberikan kontribusi hitam/putih secara individual. Struktur hasilnya dapat dideskripsikan dengan suatu $n \times m$ *Boolean* matriks $S = [s_{ij}]$, dimana $s_{ij} = 1$ jika dan hanya jika subpiksel ke-j di dalam transparansi ke-i adalah hitam. Ketika transparansi $i_1, i_2, \ldots i_r$ ditumpuk bersamaan dengan cara sedemikian rupa sehingga subpiksel tersusun dengan tepat, maka akan terlihat kombinasi *share* yang mana subpiksel hitam direpresentasikan oleh *Boolean* "*or*" dari baris $i_1, i_2, \ldots i_r$ di dalam S. Derajat keabuan dari kombinasi *share* tersebut proporsional dengan bobot Hamming $H(V)$ dari m-vektor V yang di-*or*-kan. Derajat keabuan ini diinterpretasikan oleh visual manusia sebagai hitam jika $H(V) \geq d$ dan sebagai putih jika $H(V) \leq d - \alpha m$ dengan *threshold* tetap $1 \leq d \leq m$ dan perbedaan relatif $\alpha > 0$ [1].

Di dalam skema tersebut, efek visual subpiksel hitam di salah satu transparansi tidak dapat dibatalkan oleh warna dari subpiksel di transparansi lain yang diletakkan di atasnya. Aturan ini menghasilkan teknik enkripsi yang menambahkan *noise* pada *cleartext* saat enkripsi, dan mengurangi *noise* yang sama dari *ciphertext* saat dekripsi. Selain itu juga dihasilkan model yang lebih natural dimana piksel putih direpresentasikan oleh kumpulan lengkap subpiksel putih dan piksel hitam direpresentasikan oleh kumpulan lengkap subpiksel hitam, sehingga digunakan *threshold d* dan perbedaan relatif $\alpha > 0$ untuk membedakan warna.

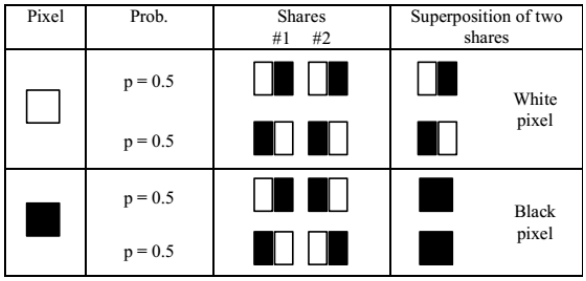
Solusi untuk skema (*k, n*) terdiri dari dua *n x m* matriks *Boolean* C_0 dan C_1. Piksel putih secara acak dipilih satu dari matriks C_0, dan piksel hitam secara acak dipilih satu dari matriks C_1. Matriks yang dipilih mendefinisikan warna dari *m* subpiksel di dalam masing-masing satu dari *n* transparansi. Solusi dinyatakan valid jika ketiga kondisi berikut terpenuhi: (1) Untuk sembarang *S* di dalam C_0, "*or" V* dari sembarang *k* dari *n* baris memenuhi $H(V) \leq d - \alpha m$, (2) Untuk sembarang *S* di dalam C_1, "*or*" *V* dari sembarang *k* dari *n* baris memenuhi $H(V) \leq d$, dan (3) Untuk sembarang subset $\{i_1, i_2, \ldots i_q\}$ dari $\{1, 2, \ldots, n\}$ dengan $q < k$, dua kumpulan $q \times m$ matriks *Dt* untuk $t \in \{0,1\}$ yang diperoleh dengan membatasi tiap-tiap $n \times m$ matriks di dalam C_1 (dimana $t = 0,1$) terhadap baris $i_1, i_2, \ldots i_q$ tidak dapat dibedakan, dalam pengertian bahwa keduanya mengandung matriks yang sama dengan frekuensi yang sama. Kondisi ketiga mengimplikasikan bahwa dengan menggunakan jumlah *share* < *k*, tidak akan dapat menentukan apakah piksel tersebut berwarna hitam atau putih [1].

Permasalahan dengan skema (2, *n*) dapat diselesaikan dengan kumpulan *n x n* matriks berikut:

C_0 = {semua matriks yang diperoleh dengan permutasi kolom dari $\begin{bmatrix} 100 & \cdots & 0 \\ 100 & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 100 & \cdots & 0 \end{bmatrix}$}

C_1 = {semua matriks yang diperoleh dengan permutasi kolom dari $\begin{bmatrix} 100 & \cdots & 0 \\ 010 & \ldots & 0 \\ \ldots & \ldots & \ldots \\ 000 & \cdots & 1 \end{bmatrix}$}

Metode yang diajukan ini menggunakan skema gambar hitam-putih dengan resolusi biner yang hanya terdiri dari piksel 0 dan 1, dimana piksel 0 artinya putih dan piksel 1 artinya hitam. Matriks yang digunakan adalah matriks berukuran 2 x 2 untuk setiap piksel dari gambar. Satu piksel tunggal akan memiliki 2 matriks. Salah satu matriks akan dipilih secara acak dan matriks yang lain akan dibangkitkan menurut warna piksel, yaitu piksel putih atau hitam. Dalam skema (2, 2) dengan ekspansi 2-subpiksel, setiap piksel tunggal dari gambar asli akan diekspansi menjadi 2-subpiksel di dalam potongan *share*. Gambar 1 berikut menunjukkan ilustrasi dari skema (2, 2) dengan ekspansi 2-subpiksel [2].



Gambar 1. Kriptografi Visual Skema (2, 2) dengan Ekspansi 2-subpiksel

Sebagaimana telah disebutkan sebelumnya bahwa terdapat banyak variasi dari kriptografi visual. Berdasarkan dari skema *secret sharing* terdapat skema *k out of k* atau (*k, k*) dan skema *k out of n* (*k, n*). Skema (*k, k*) artinya gambar asli dipecah menjadi sejumlah *k-shares* dan informasi yang tersembunyi dapat terlihat kembali dengan merekonstruksi seluruh *k-shares* tersebut. Apabila jumlah *share* yang direkonstruksi kurang dari *k* maka gambar asli tidak akan terlihat. Sedangkan skema (*k, n*) artinya gambar asli dipecah menjadi sebanyak *n-shares* dan informasi yang tersembunyi dapat terlihat kembali dengan merekonstruksi sejumlah *k-shares*, dimana nilai $k < n$ [1]. Pada paper ini skema yang digunakan adalah skema (*k, k*) dengan nilai $k = 2$.

Berdasarkan pada gambar yang diproses, terdapat 2 jenis, yaitu gambar hitam putih atau gambar berwarna. Pada metode yang diajukan ini kedua jenis gambar tersebut diterapkan.

Variasi selanjutnya adalah berdasarkan pada tipe file, yaitu *text-based*, *image-based*, dan *extended visual cryptography* yang diimplementasikan pada QR Code. Pada paper ini menggunakan ketiga jenis tipe file tersebut.

1.2. Penelitian Terkait

Pada bagian ini akan disampaikan ringkasan mengenai beberapa penelitian yang terkait dengan *framework* pembayaran elektronik dan anti-*phising*. Suatu mekanisme anti-*phising* berbasis pada kriptografi visual telah diajukan di dalam paper [3]. Pada penelitian tersebut menggunakan suatu gambar *captcha* yang dipecah dengan kriptografi visual menjadi 2 potongan share. Potongan-potongan *share* tersebut digunakan untuk proses autentikasi, dimana hanya *user* yang sah yang dapat memiliki potongan *share* yang valid.

Suatu sistem pembayaran *online* menggunakan steganografi dan kriptografi visual dipresentasikan pada paper [4]. Namun pada paper ini tidak fokus pada pendeteksian atau pencegahan *phishing*. Tidak ada

mekanisme untuk mengecek apakah suatu situs merupakan situs yang asli atau merupakan suatu situs tiruan/palsu. Partisipan yang terlibat dalam sistem yang diajukan yaitu *user*, *merchant*, bank, dan *Certified Authority* (CA). CA memiliki kesulitan untuk meneruskan informasi ke bank yang mana dan tidak ada proses validasi antara *user* dengan CA.

Pendekatan yang lain juga dipresentasikan pada paper [2], [5], dan [6], namun dalam sistem tersebut *user* dan *merchant* harus terdaftar di server bank, yang mana pada sistem yang sudah berjalan saat ini *merchant* tidak selalu memiliki kerjasama langsung dengan *card issuer*, melainkan dengan *acquirer* bank yang terkait dengan rekening miliknya.

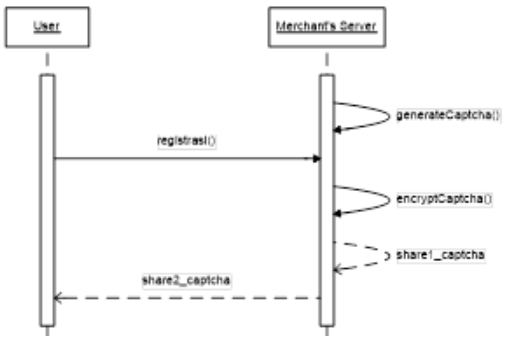
2. Metode Penelitian

Sebagai upaya untuk mendeteksi dan mencegah *phishing* dan *identity theft*, dengan tujuan untuk menghasilkan sistem pembayaran dengan kartu kredit/debit yang lebih aman, dalam artian menjamin autentikasi, otorisasi, kerahasiaan data, dan integritas data, maka dalam paper ini diajukan suatu metode pembayaran elektronik berbasis kriptografi visual. Kriptografi visual sudah banyak diterapkan di berbagai bidang, seperti *biometric*, *print online banking*, *cloud computing*, *internet voting*, dan sebagainya [7], [8]. Metode yang diajukan ini pada dasarnya mengadopsi dari sistem yang sudah ada saat ini, yaitu teknologi 3D-Secure [9], [10], sehingga *feasible* untuk diterapkan.

Pada metode yang diajukan akan terdapat 5 partisipan yang terlibat, yaitu *card holder* (selanjutnya disebut *user*), *merchant*, *acquirer*, *card issuer*, dan *card provider* dengan adanya proses validasi (autentikasi dan otorisasi) antar-partisipan yang terlibat.

Autentikasi antara *user* dengan *merchant* menggunakan gambar *captcha* yang dilakukan *image-based visual cryptography* dengan skema (2, 2) dan ekspansi 2-subpiksel [1]. Informasi pembayaran (yang dimasukkan oleh *user* di antarmuka yang disediakan oleh *Merchant Plug-In*) disembunyikan dengan kriptografi visual menggunakan *text-based visual cryptography* skema (2,2) dan penggantian piksel [11], [12]. Penyembunyian informasi pembayaran ini sekaligus juga sebagai autentikasi antara *user* dan *card provider*. Sedangkan autentikasi antara *card issuer* (melalui *Access Control Server*) dengan *user* menggunakan QR *Code* yang menyembunyikan OTP yang dilakukan *image-based visual cryptography* skema (2,2) dan ekspansi 4-subpiksel [13]. Autentikasi dan otorisasi pembayaran oleh *user* dilakukan dengan penyerahan OTP kepada *card issuer*.

Secara garis besar terdapat 3 fase utama, yaitu fase registrasi, fase login dan pembelian, dan fase pembayaran atau *checkout.* Gambar 2 berikut menunjukkan *sequence diagram* dari fase registrasi:

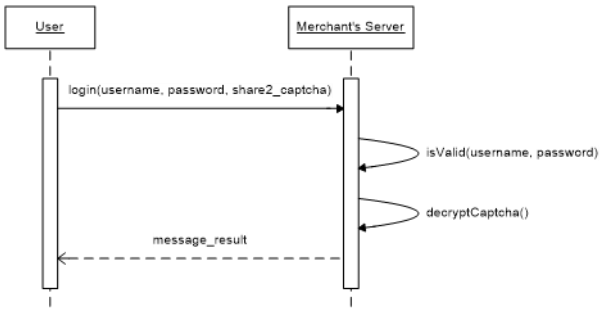


Gambar 2. *Sequence Diagram* Proses Registrasi *User*

Sebagaimana terlihat pada Gambar 2, terdapat beberapa proses yang dilakukan di fase registrasi. Pada fase ini, *merchant* membangkitkan suatu *random captcha image* yang akan tampil di halaman registrasi. Di halaman registrasi, *user* memasukkan data-data, seperti *username*, *password*, dan string *captcha* yang terlihat di layar. Jika semua masukan *user* valid, kemudian *merchant* akan membangkitkan 2 potongan *share* dari *captcha image* dengan kriptografi visual skema (2, 2). Salah satu potongan *share* disimpan di basis data *merchant*, sedangkan potongan yang lain akan dikirimkan ke *user* melalui *email.*

Fase kedua adalah login dan pembelian. Di fase login ini terdapat proses autentikasi. Proses ini dibangun sedemikian rupa sehingga sistem dapat mendeteksi apabila terdapat serangan *phishing.*

Di halaman login, *user* akan memberikan data kredensialnya berupa *username*, *password*, dan potongan *share* dari *captcha* yang diperoleh sebelumnya dari proses registrasi. Selain memverifikasi *username* dan *password*, *merchant* juga mengecek potongan *share* yang diunggah *user* dengan menumpukkannya dengan *share* yang tersimpan di basis datanya. Jika keduanya valid maka string *captcha* akan terbaca di layar. Gambar 3 berikut menunjukkan *sequence diagram* dari proses login.



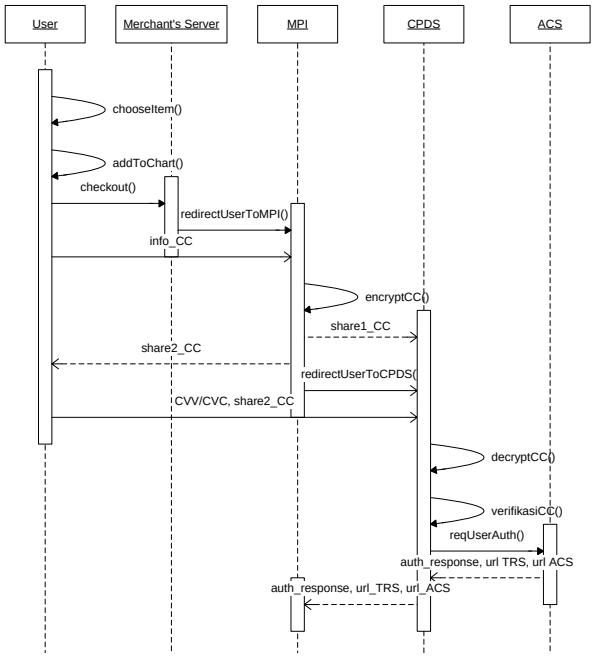
Gambar 3. *Sequence Diagram* Proses Login

Captcha dapat mendeteksi apakah *user* yang sedang login adalah *user* yang sah atau tidak, karena hanya *user* sah yang memiliki potongan *share* gambar *captcha*. Di sisi lain, apabila *share* dari *merchant* tidak valid, dalam artian situs tersebut adalah situs tiruan/palsu, maka string *captcha* juga tidak akan

terbaca, sehingga *user* dapat mengetahui bahwa situs tersebut palsu. Mekanisme login ini dapat mengidentifikasi aktivitas *phishing* dan mencegahnya secara efisien dengan 100% *true positive*.

Setelah login, maka *user* dapat memilih item yang ingin dibeli, menambahkannya ke keranjang belanja, mengisi detil pemesanan, dan melakukan pembayaran. Fase pembayaran dibagi menjadi 2 sub-fase, yaitu *checkout* dan otorisasi. Gambar 4 dan Gambar 5 berikut menunjukkan *sequence diagram* fase pembelian dan *checkout.*

Di Gambar 4 *user* berinteraksi dengan server *merchant*, *Merchant Plug-In* (MPI), *Card Provider Directory Server* (CPDS), dan *Access Control Server* (ACS). MPI adalah modul perangkat lunak yang terintegrasi dengan situs *merchant*, digunakan sebagai antarmuka antara program dari *card provider* dan program yang memproses pembayaran milik *merchant.* CPDS adalah server yang dioperasikan oleh *card provider* untuk menyalurkan *request* autentikasi dari *merchant* ke ACS milik *card issuer* dan mengembalikan hasil autentikasinya. ACS adalah server yang dioperasikan oleh *card provider* yang menyimpan daftar akun yang terdaftar dan informasi akses. ACS bertugas untuk memvalidasi *user*, memverifikai *user* pada saat pembelian, dan memberikan respon ke *merchant* [9].



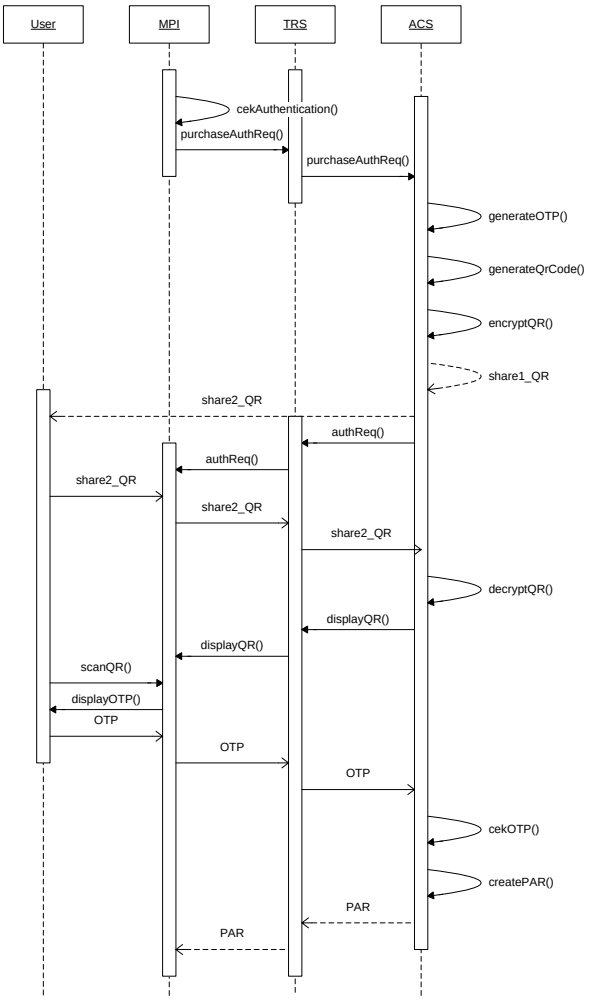
Gambar 4. *Sequence Diagram* Fase Pembelian dan *Checkout*

Pada saat *checkout* di server *merchant*, *user* diminta untuk memasukkan data kartu kredit, seperti nama yang tertera, nomor kartu, dan tanggal masa berlakunya kartu. Di fase ini *user* tidak perlu memberikan CVC/CVV. Setelah masukan disubmit, MPI akan membangkitkan 2 *share* dengan kriptografi visual dari informasi kartu tersebut. Satu *share* dikirim ke *user* dan *share* lainnya dikirim ke CPDS. *User* kemudian diteruskan ke CPDS.

Di portal CPDS, langkah selanjutnya adalah verifikasi oleh *card provider*. *User* memberikan *share*-nya dan 3 digit angka CVC/CVV. Dengan potongan *share* yang dimiliki, CPDS kemudian memverifikasi *user*. Apabila hasil penumpukan kedua *share* memperlihatkan informasi detil kartu kredit/debit, maka *user* valid dan proses dapat dilanjutkan. Sebaliknya apabila informasi detil kartu debit/kredit tidak tampil, maka dianggap bahwa *user* tidak valid dan proses dibatalkan.

Apabila *user* dan informasi kartu valid, selanjutnya *request* diteruskan ke ACS yang bersesuaian untuk menentukan apakah autentikasi tersedia untuk kartu tersebut atau tidak. Respon dari ACS kemudian dikembalikan ke MPI, termasuk URL dari *Transaction Routing Service* (TRS) milik *card provider* dan ACS milik *card issuer* yang akan digunakan oleh *merchant* untuk mengirimkan *purchase authentication request.*

Di Gambar 5, *user* berinteraksi dengan MPI, TRS, dan ACS. Semua *request* dan respon dari autentikasi diproses oleh TRS milik *card provider*.



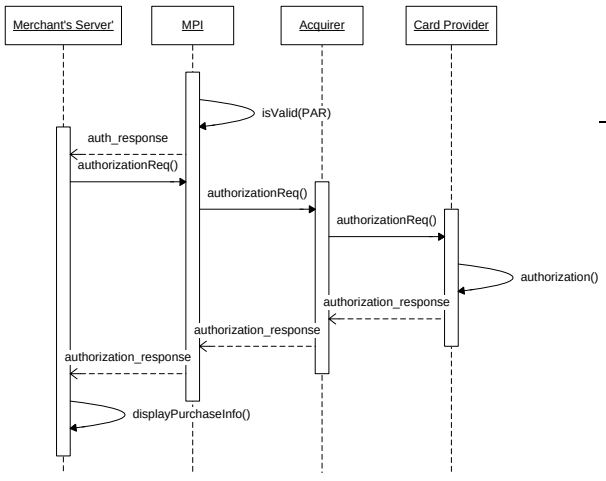
Gambar 5. *Sequence Diagram* Fase *Checkout* (lanjutan)

Setelah menerima respon autentikasi dari ACS, MPI kemudian mengecek apakah autentikasi untuk kartu tersebut tersedia atau tidak. Jika tersedia, MPI akan mengirimkan *purchase authentication request* ke ACS melalui TRS milik *card provider* via browser menggunakan link URL yag diterima dari langkah sebelumnya. *Request* tersebut berisi informasi transaksi, termasuk informasi *merchant* dan alamat URL *merchant.*

ACS kemudian membentuk *authentication request* untuk *user* dalam bentuk OTP yang tersembunyi di dalam QR Code. QR adalah teknologi yang dikembangkan Jepang yang dapat digunakan untuk menyimpan data [14]. ACS memecah QR Code tersebut menjadi 2 *share* dengan kriptografi visual. Satu *share* disimpan sementara oleh ACS dan *share* yang lain dikirimkan ke *user* melalui email. *Authentication request* dikembalikan via TRS ke browser, sehingga di jendela browser akan tampil halaman untuk memasukkan potongan *share* QR Code milik *user.*

Setelah *user* memasukkan *share*, ACS kemudian memverifikasi *user* dengan menumpuk potongan *share* tersebut dengan *share* yang disimpannya. Jika keduanya valid maka QR Code akan terbentuk dan OTP dapat diperoleh dengan memindai QR Code tersebut. Jika OTP valid, ACS membentuk *Payer Aunthentication Request* (PAR) dan mengembalikannya ke TRS yang akan meneruskan responnya ke MPI melalui browser.

MPI memvalidasi PAR dan meneruskan hasilnya ke server *merchant.* Berdasarkan data yang diterima dari MPI tersebut *merchant* akan menentukan apakah akan memprosesnya dengan otorisasi atau tidak. Fase otorisasi ditunjukkan oleh Gambar 6.



Gambar 6. *Sequence Diagram* Fase Otorisasi

Jika MPI memberikan hasil bahwa autentikasi gagal, maka *merchant* perlu meminta *user* untuk mengulangi langkah pembayaran atau menggunakan metode pembayaran yang lain.

Jika autentikasi berhasil, *merchant* kemudian mengirimkan *authorization request* ke *acquirer* yang terkait dengan rekening *merchant. Acquirer* mengirimkan *authorization request* kepada *card issuer* melalui jaringan *card provider. Card issuer* menerima dan memproses *authorization request* dan mengembalikan responnya. Jika otorisasi berhasil maka di halaman *merchant* akan tampil konfirmasi pemesanan. Sampai di tahap ini proses pembayaran dianggap selesai.

3. Implementasi

Sebuah *prototype* dikembangkan berdasarkan metode yang diajukan untuk menguji apakah metode tersebut *feasible* untuk diterapkan dan mampu meningkatkan keamanan. Pada *prototype* tersebut dibagi menjadi tiga fase utama, yaitu fase registrasi, fase login, dan fase *checkout*.

3.1. Registrasi

Pada fase registrasi di situs *merchant*, terdapat dua proses utama, yaitu pembangkitan gambar *captcha* secara random dan proses kriptografi visual untuk membagi gambar *captcha* yang telah dibangkitkan sebelumnya menjadi dua *share*, yaitu *share1_captcha* dan *share2_captcha*. *Share1_captcha* disimpan di basis data *merchant*, sedangkan *share2_captcha* dikirim kepada *user*. Adapun algoritma pada fase registrasi dapat dilihat pada Tabel 1:

Tabel 1. Algoritma Fase Registrasi

Phase	Registration
Initial State	Username, password, email, phone, captcha, etc.
Final State	Registration success, shares of captcha
Algorithm	IF isValid (username, password, email, phone, etc.) and isTrue (captcha) THEN Registration_success SEND share1_captcha to database SAVE share2_captcha to user ELSE Registration_failed

3.2. Login

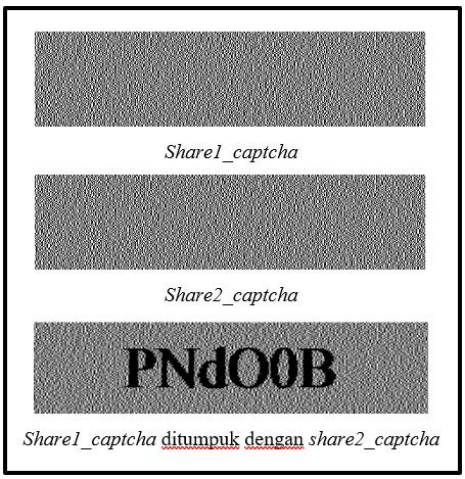
Pada fase login, *user* akan memasukkan *username*, *password*, dan *share2_captcha* yang diperoleh dari fase registrasi. Sistem akan mengecek kesesuaian *username*, *password*, dan juga *share2_captcha* dengan cara ditumpuk bersamaan dengan *share1_captcha* yang tersimpan di basis data *merchant*. Jika tulisan captcha bisa terbaca, maka secara visual dapat dikatakan bahwa *user* tersebut valid. *User* kemudian diminta untuk memasukkan string *captcha* yang dihasilkan dari hasil penumpukan *share1_captcha* dan *share2_captcha* tersebut. Sebaliknya, apabila potongan *share* yang dimasukkan oleh *user* tidak valid, maka ketika *share* tersebut ditumpuk dengan *share1_captcha* dari basis data *merchant*, string *captcha* tidak akan terlihat dan

tentu saja *user* tersebut tidak akan bisa login ke sistem. Adapun algoritma pada fase login dapat dilihat pada Tabel 2 berikut:

Tabel 2. Algoritma Fase Login

Phase	Login
Initial State	Username, Password
Final State	Username is valid, password is valid, username is invalid, password is invalid
Algorithm	IF isTrue (username, password) and isValid(captcha) THEN Username_valid ELSE Username_invalid, Password_invalid

Gambar 7 berikut menunjukkan contoh potongan *share* gambar *captcha* dan hasil penumpukannya.



Gambar 7. Potongan *Share* dari Gambar *Captcha*

3.3. *Checkout*

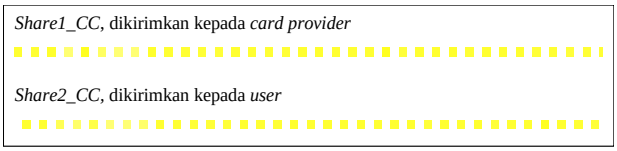
Setelah *user* berhasil masuk ke sistem *merchant*, ia dapat melakukan pembelian barang. Pada saat akan melakukan pembayaran menggunakan kartu kredit/debit atau disebut fase *checkout*, *user* diminta untuk mengisi informasi detail pembayaran dan informasi kartu kredit/debit, antara lain nomor kartu, nama yang tertera di kartu, serta tanggal masa berlaku kartu. MPI kemudian akan menjalankan kriptografi visual berbasis teks menggunakan skema (2, 2) dengan penggantian piksel menghasilkan 2 potongan *share* [11], yaitu *share1_CC* dan *share2_CC*. Langkah-langkah enkripsi dari teks menjadi *share* gambar tersebut dijelaskan pada algoritma berikut:

Algoritma 1. Enkripsi teks menjadi *image shares*

Input: file teks
Output: *image shares*

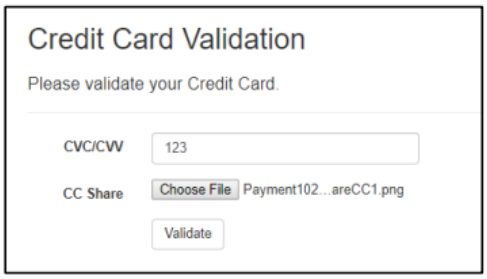
1. Baca teks yang berisi informasi kartu kredit *user*
2. Bangkitkan 2 *random noisy image shares* (*share1_CC*, *share2_CC*) dalam format PNG
3. Pindahkan semua informasi yang dibaca ke dalam *buffer*
4. Hitung *height* dan *width* dari *image shares*
   Width = jumlah karakter dalam satu baris
   Height = jumlah baris dalam satu file
5. Baca semua baris dari konten di *buffer*
6. Begin
7. Pilih tiap karakter dari baris
8. Hitung nilai ASCII dari karakter
9. Hitung piksel individual dengan SetRGB
10. Tempatkan setiap piksel dari satu baris ke dalam *share1_CC* dan *share2_CC* secara bergantian
11. Simpan piksel pada *image shares* berdasarkan pada koordinat x, y yang mana menunjukkan posisi dari karakter di dalam satu baris dan nomor baris masing-masing
12. Ulangi proses ini sampai *end of file*
13. *End*
14. Simpan *image shares* (*share1_CC* dan *share2_CC*) dalam format PNG

Gambar 8 berikut menunjukkan contoh *image shares* dari informasi kartu kredit. *Image shares* berupa piksel-piksel gambar sebagaimana contoh berikut.



Gambar 8. Contoh *image shares* hasil kriptografi visual pada informasi kartu kredit

Selanjutnya MPI akan mengirimkan *share1_CC* ke CPDS dan *share2_CC* kepada *user*. MPI kemudian mengarahkan *user* ke CPDS untuk melakukan validasi kartu kredit/debitnya dengan memasukkan 3 angka CVC/CVV yang tertera di kartu. Gambar 9 berikut menunjukkan layer validasi kartu kredit/debit.



Gambar 9. Layar Validasi Kartu Kredit

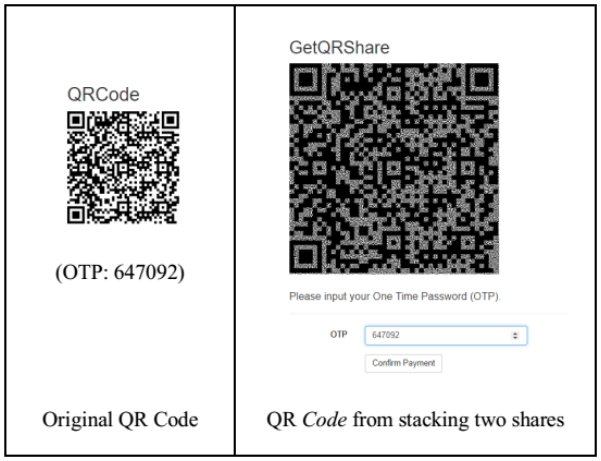
Di layer tersebut, selain memasukkan 3 angka CVC/CVV *user* juga mengunggah *share2_CC* yang dikirimkan oleh MPI. CPDS kemudian menumpuk *share2_CC* tersebut dengan *share1_CC* yang dimilikinya Jika kedua *share* valid maka CPDS akan dapat membaca informasi detil kartu kredit/debit dengan menggunakan algoritma berikut:

Algoritma 2. Dekripsi *image shares* menjadi teks

Input: *image shares*
Output: file teks

1. Baca *image shares* yang telah ditumpuk
2. Inisialisasi *string buffer*
3. Pilih setiap dan semua piksel dari baris
4. Temukan nilai ASCII dari piksel yang terpilih
5. Temukan karakter yang sesuai dengan nilai ASCII, tempatkan di dalam *buffer*
6. Tulis ulang semua data dari *buffer* ke teks
7. Baca informasi yang ada di dalam teks

Setelah informasi kartu berhasil dibaca, CPDS kemudian melakukan verifikasi kartu kredit/debit tersebut dan meneruskan *request* ke ACS dari *card issuer* yang bersesuaian untuk menentukan apakah autentikasi tersedia untuk kartu tersebut. Respon dari ACS kemudian dikembalikan ke MPI. Jika autentikasi tersedia, MPI akan mengirimkan *purchase authentication request* ke ACS. ACS kemudian membuat *authentication request* ke *user* dalam bentuk QR Code yang berisi OTP. ACS membangkitkan 2 potongan *share*, yaitu *share1_QR* dan *share2_QR*, dari QR Code menggunakan kriptografi visual skema (2, 2) dengan ekspansi 4-subpiksel [13]. *Share1_QR* dikirimkan ke *merchant* dan *share2_QR* dikirimkan ke *user*. Gambar 10 menunjukkan contoh QR Code yang dibangkitkan oleh ACS dan QR Code hasil dari rekonstruksi potongan *share*.



Gambar 10. Contoh QR Code asli dan QR Code hasil rekonstruksi kriptografi visual

Tampilan yang terlihat oleh *user* di *browser* pada saat *authentication request* adalah setelah menyerahkan CVV/CVC dan *share2_CC*, *user* melihat layar yang menampilkan informasi detail pembelian dan meminta *user* untuk menyerahkan *share2_QR*. *Share2_QR* kemudian dikombinasikan dengan *share1_QR* yang dimiliki ACS. Jika kedua *share* valid, maka akan nampak QR *Code*. *User* kemudian memindai QR Code yang berisi OTP dan menggunakannya untuk autentikasi pembayaran. Jika OTP valid maka ACS akan mengembalikan pesan bahwa autentikasi berhasil, membentuk *Payer Authentication Request* (PAR), dan mengembalikannya ke MPI untuk kemudian diteruskan ke server *merchant.*

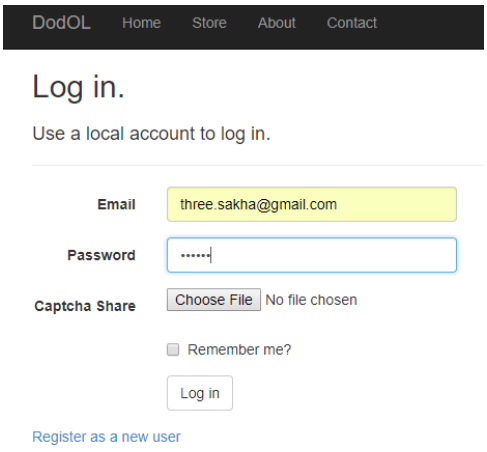
Server *merchant* kemudian memproses autorisasi dengan mengirimkan *authorization request* ke pihak *acquirer* yang bersesuaian. *Acquirer* kemudian meneruskan *request* tersebut ke *card issuer* melalui jaringan *card provider*. *Card issuer* akan memproses otorisasi tersebut dan mengembalikan respon. *Card issuer* dapat menerima atau menolak permintaan otorisasi dengan alasan yang tidak terkait dengan hasil autentikasi, misalnya saldo atau limit tidak cukup, rekening tidak aktif, dan sebagainya. Jika *card issuer* mengotorisasi transaksi maka pembayaran berhasil dilakukan. Adapun algoritma pada fase *checkout* dapat dilihat pada Tabel 3 berikut:

Tabel 2. Algoritma Fase *Checkout*

Phase	Checkout
Initial State	Order_id, User_id, CC_information
Final State	Payment_success, Payment_failed
Algorithm	IF isValid (user) and isEnough (CC limit) THEN Payment_success ELSE Payment_failed

4. Hasil dan Pembahasan

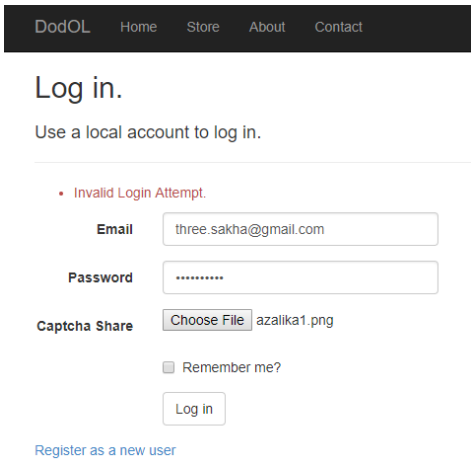
Serangkaian pengujian dilakukan terhadap *prototype* yang dibuat untuk menguji metode ini. Pengujian meliputi pengubahan informasi asli dan pengunggahan potongan *share* yang tidak bersesuaian. Percobaan dilakukan sebanyak 100 kali dan kesemuanya memberikan hasil 100% *true positif*. Gambar 11 menunjukkan salah satu hasil pengujian dengan input *password* yang tidak valid dan tanpa mengunggah *share* gambar *captcha.*



Gambar 11. Hasil Pengujian *Invalid Input*

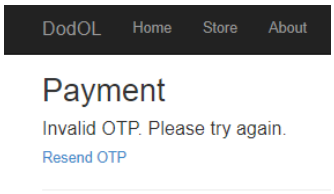
Gambar 12 berikut menunjukkan salah satu hasil pengujian dengan mengunggah *share* gambar *captcha* yang tidak bersesuaian. Hal ini mengindikasikan bahwa ada pihak yang tidak berhak (*attacker*) yang berusaha masuk ke sistem menggunakan akun *user*, namun *attacker* tersebut tidak memiliki potongan *share* gambar *captcha* yang bersesuaian.

Gambar 13 dan 14 berikut menunjukkan salah satu hasil pengujian *user* memasukkan kode OTP yang tidak valid, dimana kode OTP yang seharusnya adalah 416878 namun *user* memasukkan angka 416877. Pemasukan kode OTP yang tidak valid ini bisa dikarenakan oleh kesalahan *user* maupun karena ada *attacker* yang mencoba mengotorisasi transaksi. Namun karena *attacker* tersebut tidak memiliki potongan *share* gambar QR *Code* maka ia tidak dapat menampilkan QR *Code* yang mengandung OTP.



Gambar 12. Hasil Pengujian *Invalid Captcha Share*

Gambar 13. Pengujian Memasukkan *Invalid* OTP



Gambar 14. Hasil Pengujian *Invalid* OTP

Tujuan dari *phishing* adalah untuk mendapatkan data-data kredensial *user*, misalnya data kartu kredit/debit. Ketika *user* menerima email palsu yang meminta konfirmasi data kartu kredit/debit dengan meng-klik suatu tautan yang mengarahkan *user* ke suatu situs palsu yang menyerupai situs asli, *user* perlu memasukkan potongan *share* dari gambar *captcha* miliknya. Jika setelah memasukkan *share* tersebut string *captcha* tidak terlihat, maka *user* tidak perlu memberikan informasi kartunya, karena dapat dipastikan bahwa situs tersebut palsu. Selanjutnya *user* dapat mengunjungi situs asli dan meminta penggantian *captcha* yang baru, karena *share* dari *captcha* lama sudah dikirimkan ke situs palsu. Pada dasarnya scenario ini seharusnya tidak akan pernah terjadi, karena *merchant* tidak diperbolehkan untuk menyimpan data detil kartu kredit/debit, sehingga tidak perlu adanya konfirmasi data.

Jika dengan berbagai cara pada akhirnya *phisher* berhasil mendapatkan informasi detil kartu kredit/debit dan menggunakannya untuk bertransaksi pada situs *merchant*, maka agar transaksi pembayaran berhasil *phisher* tersebut masih membutuhkan akses ke alamat *email* dan nomor telepon *user* yang terdaftar di *card issuer* yang terasosiasi dengan akun kartu kredit/debit tersebut.

User dimungkinkan untuk melakukan pembayaran tanpa melakukan registrasi di situs *merchant.* Hal ini dapat terjadi karena data yang diperlukan untuk mengonfirmasi pembayaran dengan kartu kredit/debit hanyalah data yang terkait dengan akun kartu kredit, yang mana data tersebut tersimpan di dalam basis data *card provider* dan *card issuer*, tidak secara langsung terkait dengan data registrasi *user* di situs *merchant.*

4.1. Kelebihan

Kelebihan dari metode yang diajukan ini adalah autentikasi, otorisasi, kerahasiaan data *user*, dan integritas data lebih terjamin keamanannya.

Auntentikasi pembayaran lebih terjamin karena hanya partisipan yang sah yang memiliki potongan *share* pada setiap transaksi. *Share captcha* hanya dimiliki oleh *user* dan *merchant*, penyerang tidak akan bisa menggunakan akun *user* sekalipun memiliki *username* dan *password* karena tidak memiliki potongan *share captcha* yang dibangkitkan oleh situs *merchant. Share* data kartu kredit/debit hanya dimiliki oleh *user* dan *card provider*, penyerang tidak akan bisa meraih apapun meski berhasil mendapatkan potongan *share* kartu tersebut karena adanya proses validasi dengan memasukkan CVC/CVV yang dilakukan di portal *card provider*. *Share* QR Code hanya dimiliki oleh *user* dan *merchant.*

Otorisasi pembayaran lebih terjamin dengan adanya OTP yang disimpan di dalam QR Code yang akan diminta setiap kali terjadi transaksi dengan kartu kredit/debit, sehingga meminimalisir adanya kemungkinan transaksi yang dilakukan oleh pihak yang tidak berhak.

Kerahasiaan data *user* menjadi lebih terjamin karena adanya proses kriptografi visual yang menyembunyikan informasi kartu kredit pada saat terjadi komunikasi antar-jaringan. Selain itu tidak ada data kartu kredit yang disimpan di basis data *merchant*, sebagaimana aturan dari *PCI Data Security Standard* [15] sehingga aman dari risiko penyalahgunaan dan/atau pencurian

data, baik dari pihak internal *merchant* maupun pihak eksternal.

Integritas data lebih terjamin karena data yang dienkripsi dengan kriptografi visual tidak akan dapat diubah oleh orang yang tidak berhak. Apabila *share* hasil enkripsi diubah sedikit saja maka pada saat rekonstuksi gambar/tulisan tidak akan terlihat.

4.2. Kelemahan

Selain kelebihan yang telah disebutkan, metode yang diajukan di paper ini juga masih memiliki kelemahan, yaitu kenyamanan *user* mungkin akan berkurang dan juga *load* dari server menjadi bertambah.

Kenyamanan *user* berkurang karena proses pembayaran yang menjadi lebih panjang. Di setiap kali transaksi, *user* diharuskan untuk memasukkan data kartu kredit/debit di situs *merchant* dan CVC/CVV di portal *card provider*. *User* juga diharuskan untuk menyimpan baik-baik potongan *share* yang dimilikinya. Selain itu, agar dapat melakukan transaksi ini, *user* juga perlu memiliki dua perangkat untuk bertransaksi, dimana satu perangkat digunakan untuk proses jual-beli dan perangkat yang lainnya digunakan untuk memindai QR Code. Salah satu dari perangkat tersebut harus memiliki kamera atau pemindai untuk memindai QR Code.

Selain kenyaman *user* yang berkurang, server dari *merchant*, *card provider*, dan *card issuer* juga akan menjadi lebih sibuk. Hal ini dikarenakan adanya proses kriptografi visual pada *merchant* dan *card issuer*, proses validasi kartu kredit di *card provider*, serta pembangkitan OTP dan QR Code di *card issuer*.

5. Kesimpulan

Phishing adalah salah satu jenis serangan yang sering menjadi ancaman bagi sistem jual-beli secara daring. Metode pembayaran elektronik dengan kartu kredit/debit pada sistem jual-beli *online* yang diajukan di paper ini berbasis pada kriptografi visual. Kriptografi visual diterapkan pada gambar *captcha* di fase registrasi, serta pada informasi detil kartu kredit dan QR Code yang mengandung OTP di tahap pembayaran. Hasil pengujian menunjukkan bahwa metode yang diajukan ini dapat mencegah *phishing* dan *identity theft*, dalam artian bahwa autentikasi, otorisasi, kerahasiaan data *user*, dan integritas data menjadi lebih terjamin. *Phishing* dapat dicegah karena hanya *user* dan *merchant* yang sah yang memiliki potongan *image share*. Transaksi pembayaran dengan kartu kredit/debit oleh *user* yang tidak sah dapat dicegah karena hanya partisipan yang sah yang dapat mengonfirmasi pembayaran menggunakan OTP yang terkandung di dalam QR Code, dimana potongan *share* dari QR Code tersebut hanya akan dikirimkan ke email *user* yang terintegrasi dengan akun kartu kredit. Pencurian/penyalahgunaan data kartu kredit dapat dicegah karena data tersebut tidak tersimpan di basis data *merchant*.

Daftar Rujukan

[1] Naor M. and Shamir A., 1995. Visual Cryptography. *EUROCYPT'94*, 950, pp.1-12.
[2] Chaudari N. and Parate P., 2016. Secure Online Payment System using Visual Cryptography. *International Journal of Advanced Research in Computer and Communication Engineering*, 5(2), pp.552-553.
[3] James D. and Philip M., 2012. A Novel Anti-Phising Framework Based on Visual Cryptography. In: PES University, *2012 International Conference on Power, Signals, Controls and Computation (EPSCICON)*. Thrissur, Kerala, India 3-6 Jan 2012. IEEE.
[4] Roy S. and Ventakeswaran P., 2014. Online Payment System using Steganography and Visual Cryptography. In: Maulana Azad National Institute of Technology Bhopal, *2014 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. Bhopal, India 1-2 March 2014. IEEE.
[5] Jain N.R., Ujwal K., Apsara S., Nikhil P., and Tejashri D., 2016. Advance Phising Detection using Visual Cryptography and One Time Password. *International Journal of Advanced Research in Science, Engineering and Technology*, 3(4), pp.1808-1812.
[6] Akolkar S., Kokulwar Y., Neharkar A., and Pawar D., 2016. Secure Payment System using Steganography and Visual Cryptography. *International Journal of Computing and Technology*, 3(1), pp.58-61.
[7] Thomas, S.A., dan Gharge, S., 2017. Review on Various Visual Cryptography Schemes. In: *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*. Mysore, India 6 Sept 2018. IEEE.
[8] Jain, A. dan Soni, S., 2017. Visual Cryptography and Image Processing Based Approach for Secure Transactions in Banking Sector. *In: 2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*. Noida, India 23 April 2018. IEEE.
[9] VISA, 2011. *Verified by Visa: Acquirer and merchant implementation guide*. U.S. Region.
[10] VISA, 2019. *PSD2 SCA for Remote Electronic Transactions: Implementation Guide Version 1.1*.
[11] Brindha K. and Jeyanthi N., 2017. Securing Portable Document Format File Using Extended Visual Cryptography to Protect Data Storage. *International Journal of Network Security*, 19(5), pp.684-693.
[12] Fang W.P., Hsu J.H., and Cheng W., 2013. Text-based Visual Secret Sharing. *International Journal of Computer Science and Network Security*, 13(5), pp.38-40.
[13] Cao X., Feng L., Cao P., and Hu J., 2016. Secure QR Code Scheme Based on Visual Cryptography. In: Sehiemy R.E., Reaz M.B.I., and Lee C.J., *2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE)*. Beijing, China 20-21 Nov 2016. Atlantis Press.
[14] Tiwari, S., 2016. An Introduction to QR Code Technology. *In 2016 International Conference on Information Technology (ICIT)*. Bhubaneswar, India 22-24 Dec 2016. IEEE.
[15] PCI Security Standards Council, LLC., 2016. *Payment Card Industry (PCI) Data Security Standard: Requirement and security assessment procedures v3.2*.