Pengembangan Sistem E-Ticketing Berbasis QR Code untuk Event Pertunjukan dengan Implementasi Algoritma Schmidt-Samoa

Michel Vito Adinugroho - 18220066 (*Author*)
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: vitoadinugroho2207@gmail.com

***Abstract*— Teknologi *electronic ticket* atau yang biasa dikenal *e-ticket* telah menawarkan berbagai kemudahan penggunaan dan memberikan solusi untuk mengatasi berbagai kelemahan sistem tiket konvensional dalam aspek keamanan. Walaupun demikian, penggunaan *e-ticket* terutama pada *event* pertunjukan masih mengalami berbagai masalah berkaitan dengan pencurian data tiket, pemalsuan tiket, dan kesulitan menjamin kepemilikan tiket. Untuk mengatasi permasalahan tersebut, diterapkan konsep *digital signature* dengan algoritma Schmidt-Samoa untuk mengembangkan sistem *e-ticketing* berbasis QR Code untuk memastikan keaslian tiket, mencegah pencurian data tiket, serta menjamin kepemilikan tiket. Hasil pengujian dengan berbagai skenario uji menunjukan sistem berfungsi secara baik dalam menangani ketiga permasalahan. Selain itu, pengujian dengan panjang kunci 1024-1530 bit memberikan informasi waktu tambahan pemrosesan tiket yang ideal yakni hanya sebesar 0.06 detik untuk mempertahankan proses verifikasi tiket yang efisien.**

Kata Kunci—e-ticket; algoritma kriptografi, tanda-tangan digital, algoritma schmidt-samoa, QR Code

I. PENDAHULUAN

Perkembangan teknologi era digital mempermudah manusia dalam memperoleh tiket dengan adanya tiket digital (*electronic ticket* / *e-ticket*). Dengan tiket terdigitalisasi, calon pembeli tidak perlu khawatir tiketnya tertinggal, hilang, atau rusak [1]. E-ticket juga digunakan untuk mengatasi kelemahan proses validasi tiket yang kerap terjadi pada sistem tiket konvensional [2].

Walaupun demikian, penggunaan *e-ticket* terutama pada *event* pertunjukan masih menghadapi berbagai masalah keamanan. Pertama, pembobolan *database* dan pencurian data *e-ticket* menjadi isu di berbagai kasus, contohnya pembobolan Ticketmaster UK Limited pada tahun 2018 yang telah mengungkap informasi pribadi termasuk nama, email, nomor telepon, dan kartu kredit menyebabkan potensi penipuan bagi *customer* mereka [3]. Kedua, pembobolan data e-ticket seringkali dieksploitasi untuk pembuatan tiket palsu yang lolos pemeriksaan sehingga pemilik tiket maupun pemalsu tiket dapat menikmati pertunjukan, seringkali terjadi pada pertunjukan sepak bola dan konser [4]. Ketiga, terdapat permasalahan terkait penjualan kembali e-ticket oleh pihak yang tidak berwenang, contohnya terdapat kasus pembobolan akun pengguna untuk penjualan tiket secara ilegal yang terjadi pada situs pembelian tiket terkemuka, StubHub [5], dan kasus pencaloan tiket Coldplay yang viral di Indonesia pada tahun 2023. Kasus tersebut menimbulkan kerugian bagi para *fans* yang harus membeli tiket dengan harga 2 hingga 10 kali lipat [6].

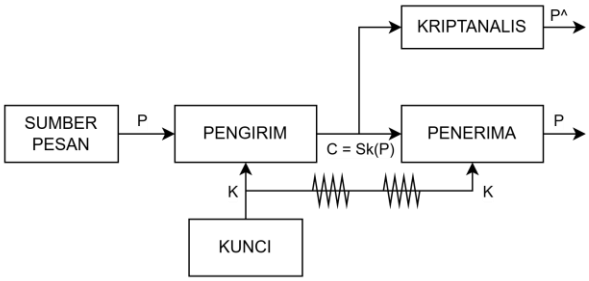
Sebagai upaya untuk mengatasi masalah pemalsuan sekaligus menjamin kepemilikan tiket agar tidak mudah dipindahtangankan, konsep *digital signature* dapat diterapkan pada *e-ticket*. Prosesnya adalah: identitas pemilik akan dienkripsi (disamarkan) di dalam *e-ticket* dengan sebuah kunci-privat yang hanya diketahui oleh pengelola *event* menjadi sebuah *signature*/cipherteks. Cipherteks ini kemudian akan di dekripsi (dikembalikan menjadi semula) menggunakan kunci-publik milik pengelola *event* yang dilakukan petugas validator tiket di lokasi *event*. Dengan ini, pembeli maupun pihak ketiga akan kesulitan untuk memalsukan tiket karena mereka tidak memiliki kunci-privat untuk menghasilkan cipherteks yang identik untuk setiap tiket. Keaslian tiket juga dapat dipastikan karena hanya tiket yang berasal dari enkripsi pengelola *event* yang dapat didekripsi. Untuk mengurangi dampak pencurian data, sistem *e-ticket* juga harus didukung oleh *database* yang terenkripsi dengan baik untuk mencegah pencurian data tiket.

Oleh karena itu, penelitian ini bertujuan untuk mengembangkan sekaligus mengevaluasi kinerja sistem *e-ticketing* yang menerapkan konsep *digital signature* pada sebuah algoritma kriptografi kunci publik pilihan dengan penyembunyian data tiket melalui media pindai. Sistem ini diharapkan dapat mengatasi kekurangan sistem *e-ticketing* yang ada, baik sisi keamanan maupun kepemilikan tiket sambil tetap menjaga kemudahan proses verifikasi tiket. Batasan masalah pada penelitian ini adalah: pengembangan sistem berfokus pada aspek keamanan, tidak mencakup ranah pembayaran, diimplementasikan sebagai purwarupa *website*, dan tidak menangani pengecekan keaslian kartu identitas pada proses validasi tiket (diasumsikan tidak ada pemalsuan kartu identitas).

II. LANDASAN TEORI

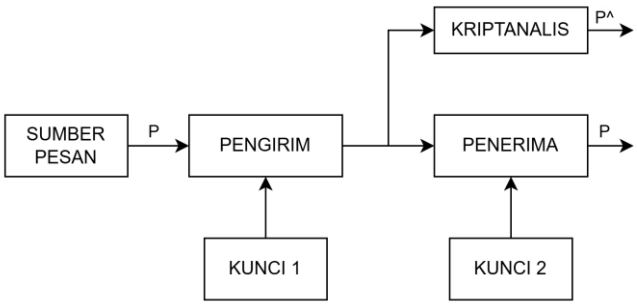
A. *Kriptografi Kunci-Publik*

Kriptografi kunci-publik merupakan penerapan sub bidang sistem kriptografi yang menerapkan penggunaan sepasang kunci yang saling invers dan sulit diturunkan satu sama lain pengiriman dan penerimaan pesan. Kriptografi kunci-publik dibuat oleh Whitfield Diffie dan Martin E.Hellman untuk mengatasi kelemahan sistem kriptografi kunci simetri yakni proses yang mengharuskan adanya distribusi kunci melalui saluran yang aman (*secure channel*) dari pengirim ke penerima pesan sebagaimana proses pada gambar 1.



Gambar 1. Sistem Kriptografi Kunci Simetris [7]

Pada kriptografi kunci-publik, kebutuhan akan distribusi kunci ditiadakan dengan penggunaan dua buah kunci; satu kunci privat (rahasia) dan satu kunci publik (boleh disebarluaskan). Dengan cara ini, semua orang dapat mengenkripsi pesan menggunakan sebuah kunci publik (kunci 1) , tetapi hanya penerima pesan yang dapat membuka pesan menggunakan kunci privat miliknya (kunci 2) [7], sebagaimana proses pada gambar 2.



Gambar 2. Sistem Kriptografi Kunci Publik [7]

Proses enkripsi dan dekripsi pesan pada kriptografi kunci publik dapat dirumuskan sebagai berikut [8]:

$$Ee(m) = c$$

$$Dd(c) = m$$

dengan E adalah fungsi enkripsi, D adalah fungsi dekripsi, (e, d) adalah pasangan kunci enkripsi dan dekripsi, c adalah cipherteks hasil enkripsi, dan m adalah pesan/plainteks yang akan dienkripsi.

B. *Algoritma RSA*

RSA adalah sebuah algoritma kriptografi yang dibuat oleh Ronald Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1978. Algoritma ini mengimplementasikan dua hal penting: kriptografi kunci publik dan tanda-tangan digital (*digital signature*) [9].

Algoritma RSA memiliki dasar keamanan dari sulitnya memfaktorkan bilangan bulat besar n menjadi faktor-faktor prima (p dan q), dalam hal ini $n = p * q$. Sebagai salah satu algoritma kriptografi kunci publik, RSA menjamin keamanan komunikasi dengan meniadakan kebutuhan akan *secure channel* untuk distribusi kunci. RSA merupakan "*trap-door one-way function*" yang memenuhi persamaan permutasi bolak balik sehingga dapat digunakan untuk mengimplementasikan "*signatures*" [9].

Penggunaan algoritma RSA melibatkan dua prosedur: prosedur pembangkitan kunci serta prosedur enkripsi dan dekripsi data. Berikut adalah tahapan pembangkitan kunci algoritma RSA [9]:

1. Menghitung sebuah nilai $n = p * q$ dari dua buah bilangan prima rahasia yang sangat besar: p & q. Nilai n bersifat publik (boleh dipublikasi) dan tidak akan mempengaruhi keamanan p dan q.
2. Memilih sebuah nilai integer besar d yang relatif prima dengan $(p-1) * (q-1)$ melalui pengecekan apakah d memenuhi gcd $(d, (p-1) * (q-1)) = 1$ ("gcd" berarti "*greatest common divisor*").
3. Menghitung nilai e sebagai "*multiplicative inverse*" dari d, yang memenuhi $e * d \equiv 1 (mod (p-1) * (q-1))$.

Melalui tahapan pembangkitan kunci tersebut, diperoleh pasangan kunci-publik (e, n) dan pasangan kunci-privat (d, n) yang digunakan dalam prosedur enkripsi dan dekripsi data dengan tahapan berikut [9]:

1. Membagi pesan M menjadi blok-blok plainteks yang berukuran integer antara 0 sampai $n-1$.
2. Melakukan enkripsi pesan dengan menghitung nilai blok cipherteks C untuk tiap blok plainteks M yang memenuhi $C = E(M) = M^e \ (mod\ n)$.
3. Cipherteks C akan dikirim ke penerima pesan. Dekripsi dilakukan untuk tiap blok cipherteks C yang memenuhi $D(C) = C^d \ (mod\ n) = M$.

Dalam penggunaan RSA sebagai algoritma pada tanda-tangan digital, RSA menjamin aspek *message-dependent*, yakni bahwa sebuah tanda-tangan digital juga ditentukan dari isi pesan, tidak hanya dari pengirim. Hal ini mencegah *copy paste* tanda-tangan sebagaimana yang terjadi pada dokumen elektronik. Prosedur tanda-tangan digital menggunakan RSA ditandai dengan pengirim yang "menandatangani (mengenkripsi)" pesan menggunakan kunci privat penerima pesan sebagai sebuah *signature* $S = D(M) = M^d \ (mod\ n)$. Penerima pesan kemudian mengekstrak *signature* dengan kunci publik milik penerima pesan sehingga didapat pesan $M = E(S) = S^e \ (mod\ n)$ [9].

C. Algoritma Schmidt-Samoa

Algoritma Schmidt-Samoa adalah algoritma kriptografi kunci-publik buatan Katja Schmidt-Samoa pada tahun 2006. Algoritma ini memiliki prosedur mirip RSA, tetapi menawarkan keamanan yang lebih baik dari RSA dari sisi matematis, yakni dengan menggunakan proses perpangkatan $N = p^2q$ dalam pembangkitan kunci yang meningkatkan jumlah kemungkinan hasil pemfaktoran [10]. Schmidt-Samoa menjelaskan algoritma ini dalam dua tahap prosedur, yakni prosedur pembangkitan kunci serta prosedur enkripsi dan dekripsi pesan. Berikut adalah tahapan pembangkitan kunci algoritma schmidt-samoa:

1. Menghitung sebuah nilai $n = p^2q$ dengan memilih dua buah bilangan prima yang sangat besar p & q. Nilai n adalah kunci-publik yang boleh dipublikasi karena nilai p dan q sulit diturunkan (difaktorkan) dari bilangan besar n.
2. Menentukan nilai d sebagai kunci-privat dengan perhitungan $d = n^{-1} \, mod \, lcm(p-1, q-1)$, dengan "lcm" adalah "*least common multiple*" atau kelipatan persekutuan terkecil.
3. Menghitung dan menyimpan nilai pq untuk digunakan dalam proses dekripsi.

Melalui tahapan pembangkitan kunci tersebut, diperoleh sebuah kunci-publik (n) dan pasangan kunci-privat (d, pq) yang digunakan dalam prosedur enkripsi dan dekripsi data dengan tahapan berikut [10]:

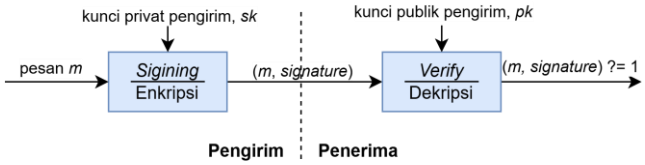
1. Mengenkripsi pesan M menjadi sebuah cipherteks c dengan perhitungan $c = m^n mod \, n$.
2. Mengirim c kepada penerima yang kemudian didekripsi kembali menjadi M dengan perhitungan $M = c^d mod \, pq$

Seperti algoritma RSA, algoritma schmidt-samoa juga dapat diimplementasikan sebagai *digital signature* (DSA-SSC) dengan menggunakan kunci-privat milik pengirim untuk mengenkripsi pesan dan menggunakan kunci-publik milik pengirim untuk mendekripsi pesan [11].

D. Tanda-tangan Digital

Tanda-tangan digital (*digital signature*) merupakan penerapan kriptografi kunci publik untuk menandatangani data digital. Tanda-tangan digital memungkinkan seorang pengirim pesan (*signer* S) untuk menandatangani pesan menggunakan kunci-privat miliknya sehingga orang yang memiliki kunci-publik milik S dapat memverifikasi bahwa pesan tersebut benar ditulis dari S (otentikasi) dan tidak dimodifikasi (anti penyangkalan) [12].

Skema tanda-tangan digital merupakan kebalikan dari skema kriptografi kunci-publik, yakni pesan ditandatangani dengan "mendekripsi" pesan menggunakan kunci-privat untuk menghasilkan signature, dan proses verifikasi dilakukan dengan "mengenkripsi" signature menggunakan kunci-publik [12]. Beberapa algoritma kriptografi yang bersifat identik (persamaan enkripsi dan dekripsinya dapat dipertukarkan) dapat digunakan untuk menandatangani pesan, salah satunya algoritma RSA dan Schmidt-Samoa (DSA-SSC).



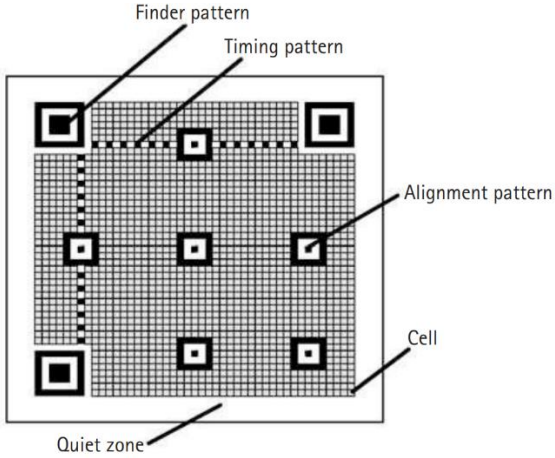
Gambar 3. Skema Tanda-tangan Digital [12]

E. Quick Response Code

Quick Response Code atau yang biasa dikenal dengan sebutan QR Code adalah kode batang dua dimensi penyimpan data yang diciptakan oleh Denso Wave. QR Code dibuat atas adanya kebutuhan penyimpanan informasi lebih yang tidak dapat ditangani oleh barcode yang hanya mampu menyimpan 20 karakter. QR code menyimpan informasi sebagai rangkaian piksel berbentuk persegi dalam kisi, dengan representasi data dalam bentuk bits (0 dan 1) [13].

QR Code yang beredar dengan berbagai versi mampu menyimpan data hingga 7089 digit angka dan 4296 karakter alphanumeric. Beberapa jenis QR Code yang baru juga diperkenalkan untuk memenuhi kebutuhan yang lebih, seperti QR Mikro untuk memenuhi kebutuhan kode yang lebih kecil dan SQRC untuk kebutuhan kriptografi dengan *public-private code*. QR Code telah disetujui sebagai standar kode internasional oleh International Organization for Standardization (ISO) pada tahun 2000 [13].

Berikut adalah struktur penyusun bagian QR Code[13].



Gambar 4. Struktur QR Code [13]

1. *Finder pattern;* yakni pola tiga buah kotak di sudut QR Code yang mengidentifikasi posisi/orientasi QR Code dan memampukan QR Code dipindai dari segala arah.
2. *Alignment Pattern;* yakni kumpulan kotak yang tersebar di permukaan QR Code, berfungsi agar QR code dapat dipindai pada permukaan yang bergelombang.

3. *Timing Pattern;* yakni kotak hitam putih yang membantu pemindai mengkonfigurasi jaringan data secara akurat dan menentukan titik tengah QR Code apabila permukaan kode terdistorsi.
4. *Quiet Zone;* yakni zona putih yang memberi ruang pembacaan dan menghindarkan kode dari objek lain saat proses *scan.*
5. *Data Area;* merupakan tempat semua data disimpan dalam tubuh QR Code. Di bagian ini terdapat *error correction block* yang mengoreksi kesalahan data yang mungkin timbul dari kerusakan permukaan QR Code.

F. Penelitian Terkait

1. Implementasi QR Code Sebagai Tiket Masuk Event dengan Memperhitungkan Tingkat Korelasi Kesalahan [2]

Penelitian ini mengimplementasikan QR Code sebagai media pindai pada e-ticket, sekaligus menguji tingkat kesalahan pembacaan QR Code yang mungkin terjadi berdasarkan beberapa level koreksi kesalahan. Penelitian ini membuktikan kemampuan QR Code sebagai media pindai e-ticket yang handal, tetapi belum memiliki mekanisme pengamanan terhadap data yang tersimpan di dalam tiket.

2. Implementasi Algoritma Schmidt-Samoa Pada Enkripsi Dekripsi Email Berbasis Android [14]

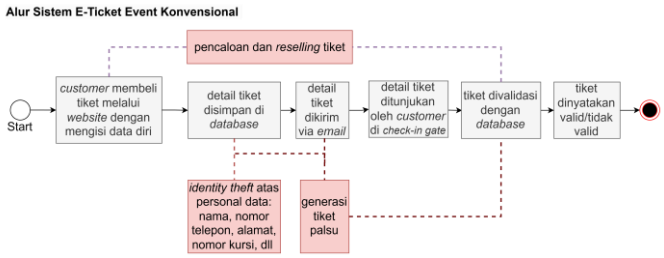
Penelitian ini mengujicobakan algoritma schmidt-samoa sebagai pengamanan pesan teks pada email menggunakan kunci berukuran 512 dan 1024 bit. Penelitian ini memberikan informasi waktu enkripsi dan dekripsi pada kedua panjang kunci tersebut, yang menyatakan waktu pemrosesan yang *applicable* untuk digunakan pada pengamanan *e-ticket*.

III. ANALISIS MASALAH & SOLUSI

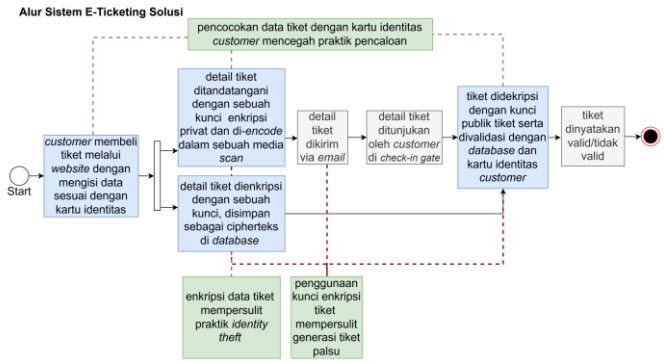
A. Analisis Masalah dan Perancangan Solusi

Berdasarkan latar belakang masalah yang dijabarkan pada pendahuluan, terdapat tiga permasalahan utama pada sistem *e-ticketing* yang bersifat konvensional, yakni pencurian data *e-ticket* [3], pemalsuan *e-ticket* [4], dan penjualan kembali *e-ticket* oleh pihak yang tak berwenang/pencaloan [5]. Berdasarkan latar belakang pula terdapat beberapa peluang yang dapat dilakukan untuk mengatasi permasalahan tersebut adalah: penerapan skema enkripsi dan dekripsi data tiket pada *database*, penggunaan media *scan* sebagai penyamaran informasi, penerapan *digital signature* untuk merahasiakan isi data tiket sekaligus menjamin keaslian tiket, dan penerapan sistem verifikasi tiket dengan adanya pencocokan identitas tiket dengan kartu identitas pemilik tiket.

Berdasarkan permasalahan dan peluang yang ada, berikut adalah perbandingan alur sistem *e-ticketing* konvensional dengan alur sistem e-ticketing solusi yang dijabarkan pada gambar 5 dan 6.



Gambar 5. Alur Sistem *E-Ticketing* Konvensional



Gambar 6. Alur Sistem *E-Ticketing* Solusi

Tabel 1. Legenda Diagram Alur Sistem

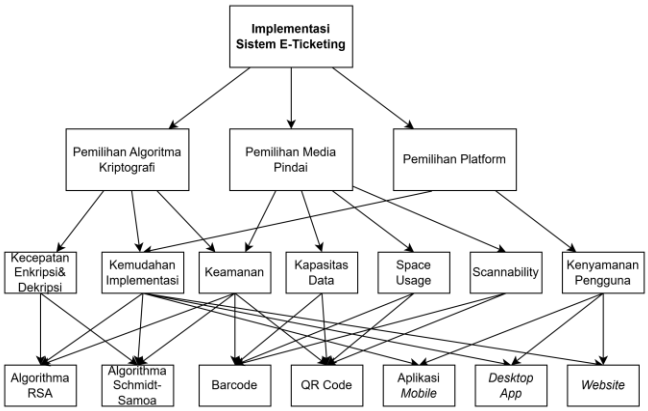
Simbol	Deskripsi
	Aktivitas konvensional yang dilakukan pada proses bisnis pembelian *e-ticket*
	Masalah yang terjadi berkaitan dengan aktivitas yang dilakukan
	Aktivitas solusi yang menggantikan aktivitas konvensional
	Dampak atas masalah akibat penggantian aktivitas konvensional menjadi aktivitas solusi
	Transisi dari suatu aktivitas ke aktivitas lain
	Percabangan yang memecah satu aktivitas menjadi dua atau lebih aktivitas yang terjadi dalam satu waktu
	Hubungan atas aktivitas yang menimbulkan masalah/dampak
Start	Titik awal dari aliran aktivitas
	Titik akhir dari aliran aktivitas

Alur kerja dari sistem solusi mengubah tiga aktivitas utama sistem konvensional yakni: *customer* kini menggunakan data diri sesuai kartu identitas miliknya untuk memastikan kepemilikan tiket; detail tiket ditandatangani dengan kunci enkripsi privat milik pengelola *event* sehingga tiket palsu akan

sulit dibuat tanpa mengetahui keberadaan kunci privat; detail tiket disimpan di *database* sebagai cipherteks dengan mengenkripsi data dengan kunci rahasia yang juga berbeda dengan kunci dekripsi *database* sehingga mempersulit pencurian data dan pembuatan tiket palsu.

B. Pemilihan Alternatif Solusi

Berdasarkan alur *e-ticketing* solusi yang telah dirancang, dilakukan pemilihan detail algoritma, media *scan*, dan platform pengembangan bagi sistem. Pemilihan solusi terbaik dilakukan dengan metode *Analytic Hierarchy Process* (AHP). Berikut adalah struktur hierarki yang digunakan dalam pemilihan detail solusi sistem pada gambar 7.



Gambar 7. Hierarki Pemilihan Solusi Sistem

Selanjutnya, berbagai alternatif solusi yang ada dibandingkan berdasarkan bobot kepentingan relatif dari masing-masing kriteria dengan pembobotan relatif dengan skala kepentingan 1-5 sebagai berikut: kecepatan enkripsi dan dekripsi (4), kemudahan implementasi (3), keamanan (5), kapasitas data (4), space usage (2), scannability (3), dan kenyamanan pengguna (4).

Tabel 2. Pemilihan Alternatif Solusi

Pemilihan Algoritma Kriptografi		
Kriteria	**Algoritma RSA**	**Algoritma Schmidt-Samoa**
Kecepatan Enkripsi & Dekripsi	9	8
Kemudahan Implementasi	8	8
Keamanan	8	9
Pemilihan Media Pindai		
Kriteria	**Barcode**	**QR Code**
Kapasitas Data	6	9
Keamanan	5	7
Space Usage	9	7
Scannability	5	9
Pemilihan Platform Implementasi		
Kriteria	**Mobile App**	**Website**	**Desktop App**
Kenyamanan	7	9	6
Kemudahan Implementasi	6	8	8

Dengan melakukan perhitungan skor agregat dengan skor pada tabel 2, didapatkan alternatif solusi yang dipilih adalah sistem *e-ticketing* yang menerapkan algoritma *schmidt-samoa* sebagai media enkripsi tiket, menggunakan QR Code sebagai media pindai, serta diimplementasikan pada *platform website*

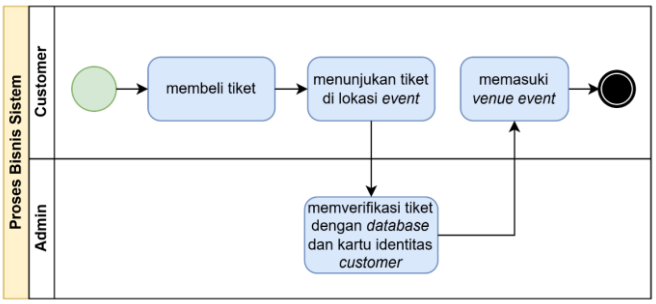
C. Desain Solusi

Bagian ini berisi pemodelan sistem berbentuk representasi visual dengan *Unified Modeling Language* (UML) yang menggambarkan interaksi antara pengguna (*customer* dan admin) dengan sistem.

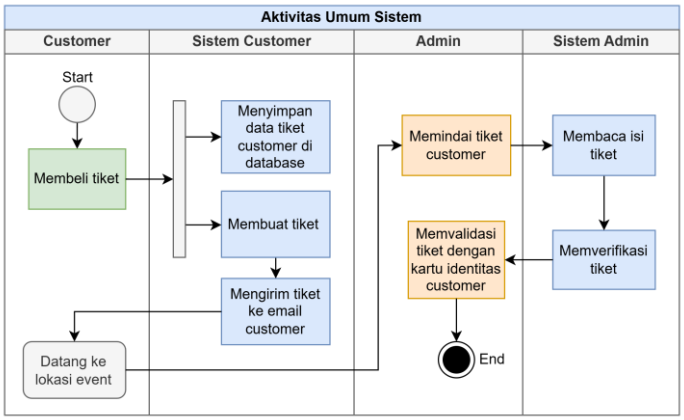
1. Proses Bisnis dan Aktivitas Sistem

Proses bisnis dimulai ketika *customer* ingin membeli tiket pada *website*. *Customer* akan mengisi data diri sesuai dengan kartu identitas serta mengisi pilihan tempat duduk yang diinginkan. Usai menekan tombol "beli", *customer* akan memperoleh tiket berbentuk QR Code melalui email. Tiket tersebut sudah ditandatangani secara digital dengan enkripsi kriptografi dan disimpan sebagai cipherteks pada *database event*.

Pada *venue event*, petugas akan melakukan verifikasi atas tiket milik *customer* dengan bantuan *website* admin. Tiket akan divalidasi oleh sistem dengan *database*, kemudian divalidasi oleh petugas dengan kartu identitas milik *customer*. Apabila tiket dinyatakan valid oleh sistem dan sesuai dengan kartu identitas, maka *customer* dapat memasuki *venue event*. Sebaliknya apabila tiket dinyatakan tidak valid oleh sistem, atau tidak sesuai dengan kartu identitas, maka tiket dinyatakan tidak valid dan *customer* tidak dapat memasuki *event*.



Gambar 8. Proses Bisnis Sistem

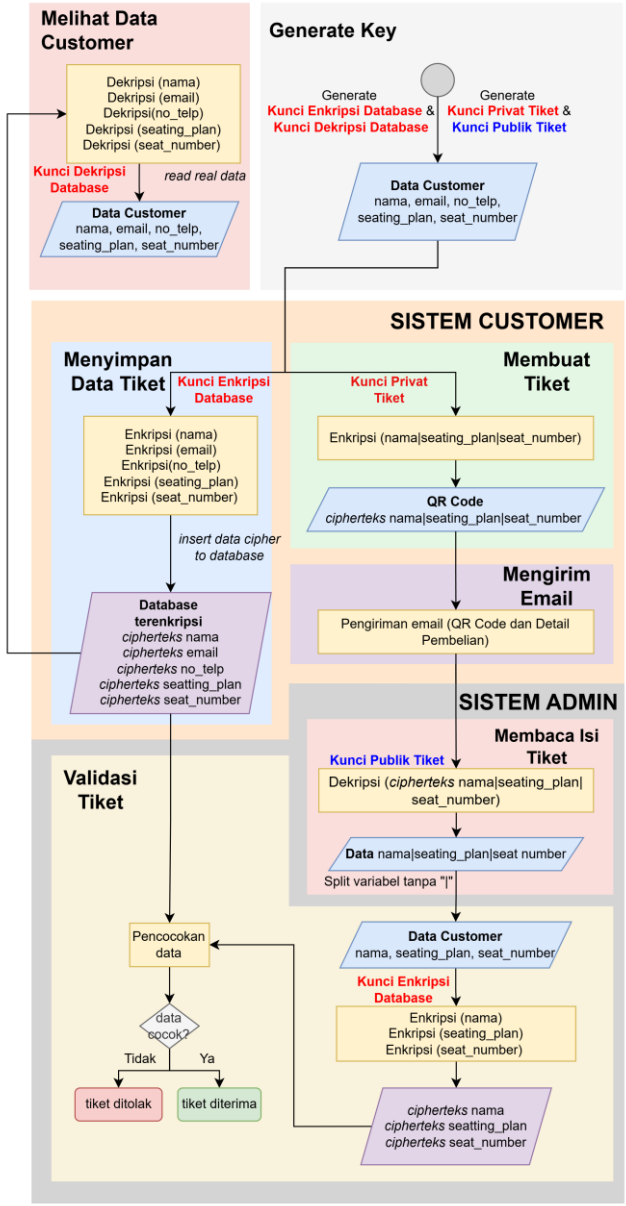


Gambar 9. Aktivitas Sistem

Aktivitas yang terjadi antar pengguna (*customer* dan admin) dengan sistem (sistem customer dan sistem admin) dijelaskan melalui gambar 9. Aktivitas dimulai dengan prosedur: *customer* membeli tiket kemudian sistem *customer* akan merespon dengan aksi dengan menyimpan data tiket *customer* di *database*, membuat tiket dalam QR Code, dan mengirim tiket ke email *customer*, lalu admin melakukan pemindaian tiket kemudian sistem admin merespon dengan aksi membaca isi tiket dan memverifikasi tiket. Proses yang terjadi pada aktivitas sistem *customer* dan sistem admin dijelaskan melalui *flowchart* sistem.

2. Flowchart Sistem

Flowchart sistem memberikan perspektif dari sudut pandang algoritma mengenai langkah-langkah penyelesaian seluruh aktivitas pada sistem *customer* dan sistem admin.



Gambar 10. Flowchart Sistem

Tabel 3. Legenda *Flowchart* Sistem

Simbol	Nama	Deskripsi
Start	*Start*	Titik awal sebuah aliran proses pada *flowchart*
	Proses	Proses yang dilakukan sistem
	Garis Alir	Arah aliran proses yang terjadi di dalam sistem
	Data	Representasi data yang digunakan atau dihasilkan oleh sebuah proses
	Data di *Database*	Representasi data yang berada di dalam tabel *database*
	Keputusan	Percabangan sebuah proses akibat adanya suatu kondisi tertentu

Detail seluruh aktivitas pada sistem *customer* dan sistem admin digambarkan melalui *flowchart* sistem pada gambar 10. Sebelum seluruh aktivitas dapat dilakukan di sistem *customer* dan sistem admin, pemilik *event* terlebih dahulu membangkitkan dua buah pasangan kunci berbeda dengan untuk digunakan menggunakan proses yang telah dijelaskan pada subbab II.3. Kunci yang dihasilkan adalah kunci publik tiket, kunci privat tiket, kunci enkripsi *database*, dan kunci dekripsi *database*. Kedua kunci yang digunakan pada database dirahasiakan dan bersifat privat. Kunci privat tiket dan kunci enkripsi *database* akan digunakan oleh sistem *customer* untuk membuat tiket dan menyimpan tiket. Sementara itu, kunci publik tiket dan kunci enkripsi *database* akan digunakan pada sistem admin untuk proses dekripsi tiket dan validasi tiket. Kunci dekripsi *database* dapat digunakan oleh pemilik *event* apabila diperlukan untuk melihat detail tiket milik semua *customer*. Kunci ini berada diluar sistem dan tidak disimpan di sistem *customer* maupun di sistem admin.

Aktivitas pada sistem *customer* dimulai ketika *customer* melakukan *checkout* pembelian tiket. Pada aksi tersebut, data *customer* berisi nama, email, nomor telepon, seating plan, dan seating number akan dihasilkan. Data *customer* kemudian digunakan sistem untuk melakukan tiga proses; menyimpan data tiket di *database* dengan mengenkripsi data *customer*; membuat tiket dengan mengenkripsi sebagian data *customer* pada QR Tiket; dan mengirim email yang berisi QR Code dan detail pembelian. Untuk penyimpanan di *database*, sistem akan mengenkripsi tiap atribut data *customer* menggunakan kunci enkripsi *database*. Hasil enkripsi tersebut adalah sebuah data cipherteks yang dimasukkan ke dalam *database*. Untuk pembuatan QR Tiket, sistem akan mengenkripsi gabungan string nama, seating plan, dan seat number milik customer dengan kunci privat tiket, dipisahkan dengan karakter pemisah *vertical bar* "|". Data tiket yang sudah terenkripsi (berbentuk

cipherteks) kemudian akan di-*encode* ke dalam QR Code dan dikirim melalui email ke *customer*.

Aktivitas pada sistem admin dimulai ketika admin memindai QR Code tiket *customer* pada *website* admin. Pada aksi tersebut, QR Code akan terlebih dahulu di-*decode* oleh sistem menjadi sebuah string cipherteks tiket. Cipherteks tersebut akan didekripsi menggunakan kunci publik tiket menjadi string data nama, seating plan, dan seating number yang dipisahkan oleh dua buah karakter pemisah vertical bar "|". Apabila hasil dekripsi tidak mengandung dua buah karakter "|" maka sistem akan menampilkan pesan bahwa tiket tidak valid. Apabila hasil dekripsi mengandung dua buah karakter "|", proses akan dilanjutkan dengan melakukan *split* atribut untuk memisahkan ketiga atribut tersebut tanpa karakter pemisah, lalu mengenkripsinya dengan kunci enkripsi *database*. Terakhir, hasil data tiket yang terenkripsi berupa cipherteks akan dicocokan dengan *row* pada *database*. Apabila ada *row* yang bersesuaian, sistem akan melakukan pengecekan kondisi atribut check-in tiket, apakah bernilai *true* (tiket sudah digunakan dan sistem menyatakan bahwa tiket tidak valid) atau *false* (tiket dinyatakan valid dan sistem akan memperbarui status check-in menjadi *true*). Jika tidak terdapat *row database* yang sesuai dengan data tiket, maka tiket langsung dinyatakan tidak valid.

IV. IMPLEMENTASI

A. Deskripsi Sistem

Sistem yang dikembangkan berdasarkan desain solusi adalah berupa sistem pembelian dan verifikasi *electronic ticket* (e-*ticket*) dengan penerapan *digital signature* dalam media pindai yang dapat meningkatkan keamanan penggunaan tiket sekaligus menjamin kepemilikan tiket. Dengan sistem ini, *customer* (pembeli tiket) dapat melakukan pembelian tiket untuk sebuah event secara mandiri melalui *website customer*, dan admin (petugas verifikator tiket) dapat melakukan verifikasi keaslian tiket customer melalui website admin.

Batasan dari sistem yang dikembangkan adalah:

1. Sistem diimplementasikan sebagai prototipe *website* secara lokal
2. Sistem tidak mencakup ranah pembayaran tiket.
3. Sistem *website* tidak menangani pengecekan keaslian kartu identitas pada proses validasi tiket (diasumsikan tidak ada pemalsuan kartu identitas).
4. Sistem yang dikembangkan belum memiliki kemampuan untuk melakukan generasi kunci baru yang secara otomatis menggantikan kunci lama.

B. Lingkungan Pengembangan Sistem

Pengembangan sistem pada penelitian ini menggunakan berbagai perangkat keras dan perangkat lunak dengan spesifikasi sebagai berikut:

1. Perangkat Keras

Perangkat keras yang digunakan selama proses pembangunan sistem adalah laptop ASUS ZenBook UM431DA dengan prosesor AMD Ryzen 5 3500U @2.1GHz dan 8 GB Memori RAM.

2. Perangkat Lunak

Perangkat lunak yang digunakan untuk membangun sistem: sistem operasi Windows 10; DBMS PostgreSQL; bahasa pemrograman python, HTML, CSS, dan JavaScript; *code editor* Visual Studio Code; dan *web framework* Django.

C. Implementasi Basis Data

Basis data diimplementasikan sebagai satu buah tabel customer_ticket yang menampung detail tiket milik *customer* yang telah diubah kedalam bentuk cipherteks. Tabel customer_ticket memiliki tujuh buah atribut data dengan penjelasan sebagai berikut:

1. **id**, merupakan *primary key* pada tabel customer_ticket. Atribut ini berfungsi untuk mengidentifikasikan setiap record secara unik dari pengguna/customer yang telah melakukan pembelian tiket. Atribut id mempunyai tipe data bigint.
2. **name**, merupakan atribut yang berfungsi untuk menyimpan data nama customer yang telah membeli tiket dalam bentuk cipherteks. Atribut name mempunyai tipe data text dan bersifat tidak null.
3. **email**, merupakan atribut yang berfungsi untuk menyimpan data email customer dalam bentuk cipherteks. Atribut email mempunyai tipe data text dan bersifat tidak null.
4. **phone**, merupakan atribut yang berfungsi untuk menyimpan data nomor telepon customer dalam bentuk cipherteks. Atribut phone mempunyai tipe data text dan bersifat tidak null.
5. **seating_plan**, merupakan atribut yang berfungsi untuk menyimpan data pilihan jenis tempat duduk customer dalam bentuk cipherteks. Atribut seating_plan mempunyai tipe data text dan bersifat tidak null.
6. **seat_number**, merupakan atribut yang berfungsi untuk menyimpan data nomor tempat duduk customer dalam bentuk cipherteks. Atribut seat_number mempunyai tipe data text dan bersifat tidak null.
7. **check-in**, merupakan atribut bertipe boolean *true* dan *false* yang berfungsi untuk menyimpan status tiket, yakni tiket yang belum melakukan check-in (false) dan tiket yang telah melakukan check-in (true).

D. Implementasi Algoritma dan Alur Kerja Sistem

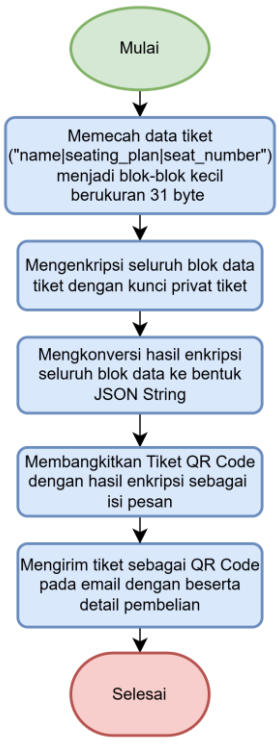
Berdasarkan desain solusi, konsep *digital signature* diterapkan pada *e-ticket* sebagai upaya untuk mengatasi masalah pemalsuan, pencurian data tiket, sekaligus menjamin kepemilikan tiket agar tidak mudah dipindahtangankan. Untuk mengatasi pemalsuan tiket, solusi yang diterapkan adalah dengan menerapkan *digital signature* dengan mengenkripsi (menyamarkan sekaligus menandatangani) identitas *customer* di dalam *e-ticket* dengan sebuah kunci-privat yang hanya diketahui oleh pengelola *event*. Hasil enkripsi identitas *customer* adalah sebuah *signature*/cipherteks yang dibenamkan di dalam QR Code. Cipherteks ini kemudian akan didekripsi (dikembalikan menjadi semula) menggunakan kunci-publik

pengelola *event* yang dilakukan petugas validator tiket di lokasi *event*. Dengan ini, pembeli maupun pihak ketiga tidak dapat memalsukan tiket apabila mereka tidak memiliki kunci-privat untuk menghasilkan cipherteks yang identik untuk setiap tiket. Keaslian tiket juga dapat dipastikan karena hanya tiket yang berasal dari enkripsi pengelola *event* lah yang dapat didekripsi.

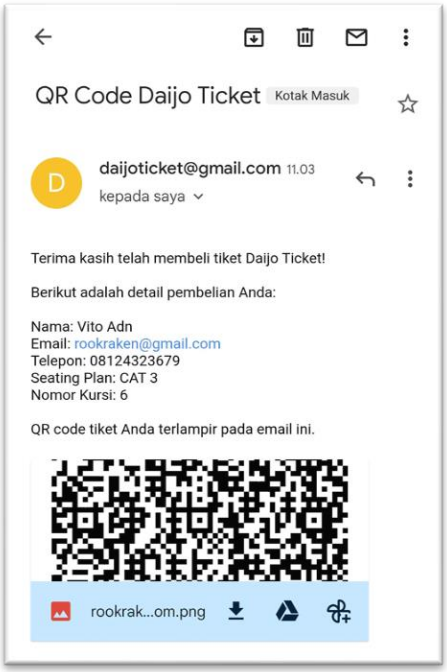
Penggunaan algoritma kriptografi bertipe kunci-publik dimaksudkan untuk diadakannya layanan validasi tiket yang lebih fleksibel pada event tanpa menurunkan tingkat keamanan. Dengan penggunaan dua buah kunci: kunci publik dan kunci privat, pengelola *event* dapat melakukan *outsourcing* petugas validator tiket ketika diperlukan (misalnya event besar) tanpa harus takut kunci yang digunakan untuk membuat tiket mengalami kebocoran (kunci untuk deskripsi bersifat publik). Sistem ini didukung pula dengan *database* yang terenkripsi untuk menghindari pencurian identitas *customer* dengan mendapatkan akses *read* ke *database*. QR Code digunakan sebagai media *e-ticket* untuk tetap menjaga kemudahan verifikasi dengan hanya melakukan pemindaian.

1) Implementasi Tiket QR Code

Implementasi tanda tangan digital pada tiket dilakukan menggunakan algoritma Schmidt-Samoa untuk mengenkripsi (menandatangani) tiket milik *customer* dengan kunci privat tiket. Proses pembuatan tiket QR Code dengan mengenkripsi string "name|seating_plan|seat_number" mengikuti prosedur yang dijelaskan melalui diagram alir pada gambar 11.



Gambar 11. Alur Pembangkitan Tiket QR Code



Gambar 12. Hasil Pengiriman Tiket via Email

Gambar 13. Detail Tiket QR Code

Gambar 12 menunjukan contoh hasil implementasi pembangkitan dan pengiriman tiket QR Code ke email customer sedangkan gambar 13 adalah contoh detail QR Code tiket yang berisi cipherteks angka: [247376551461350242612864530992985752156811561070744052090440057327007717680970836864057843448968562244374784874570179781358693326357230885824113442436283779966849947086308754341068389990802127879031396347702136311562771753065623 76]. Kode QR yang dihasilkan oleh sistem menggunakan level koreksi L yang mampu memulihkan 7% kerusakan jika QR Code mengalami kecacatan. Versi QR Code yang digunakan adalah versi 10 berukuran 57x57 piksel, berkapasitas hingga 1063 karakter alphanumerik.

2) Implementasi Enkripsi Tiket pada Database

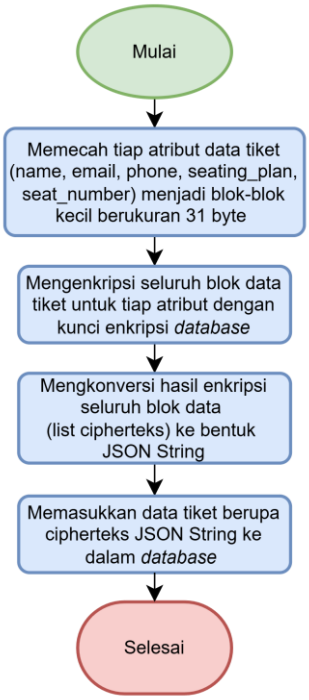
Enkripsi data tiket pada database dimaksudkan untuk mencegah pencurian identitas tiket melalui akses *read* di database. Pengamanan database dilakukan menggunakan dua buah kunci, yakni kunci enkripsi database dan kunci dekripsi database yang dibangkitkan menggunakan algoritma schmidt-samoa dengan prosedur pembangkitan kunci sesuai dengan landasan teori. Kedua kunci yang dihasilkan dibuat bersifat privat. Penggunaan kunci ganda pada sistem (walau keduanya

bersifat privat) dimaksudkan atas alasan keamanan yakni untuk mempersulit pembuatan *entry* palsu pada database.

Pada proses verifikasi tiket, sistem menggunakan kunci enkripsi *database* untuk mengenkripsi kembali sebagian data customer yang terdapat di QR Code (atribut nama, seating_plan, dan seat_number) dan melakukan pencocokan data dengan *database*. Hal ini dilakukan karena opsi untuk mendekripsi seluruh data pada *database* tiap kali melakukan validasi tiket sangatlah mahal dari segi memori dan waktu.

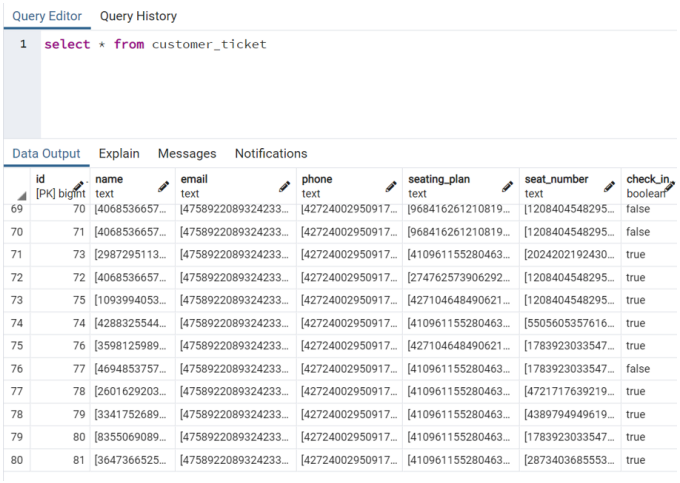
Oleh karena itu, digunakan kunci yang berbeda untuk melakukan enkripsi dan dekripsi untuk meminimalkan kemungkinan pihak ketiga membuat entri baru yang valid di *database* karena pihak ketiga tersebut tidak mengetahui format penulisan data yang digunakan. Format penulisan data hanya dapat diketahui dengan cara mendekripsi seluruh *database* dengan kunci lain (kunci dekripsi database) yang disimpan oleh penyelenggara. Kunci dekripsi *database* ini tidak digunakan dan tidak diimplementasikan dalam pengembangan sistem, tetapi dapat digunakan seandainya penyelenggara *event* ingin melihat keseluruhan isi *database* tiket yang tidak dalam kondisi terenkripsi.

Berikut merupakan alur implementasi enkripsi data tiket pada *database* yang digambarkan dengan diagram alir.



Gambar 12. Alur Enkripsi Tiket ke *Database*

Hasil penyimpanan tiket terenkripsi pada *database* ditunjukan pada gambar 13, dengan tiap isi atribut data tiket sudah berbentuk cipherteks untuk atribut name, email, phone, seating_plan, dan seating_number. Ukuran data yang dihasilkan per 1 *record* data adalah sekitar 1,5kB. Dengan demikian, perkiraan ukuran data pada berbagai *event* dengan jumlah pengunjung *event* 1000, 10 ribu, dan 100 ribu orang adalah 1.5MB, 15MB, dan 150MB yang dinilai cukup kecil dan *applicable*.


Query Editor | Query History

```
1  select * from customer_ticket
```

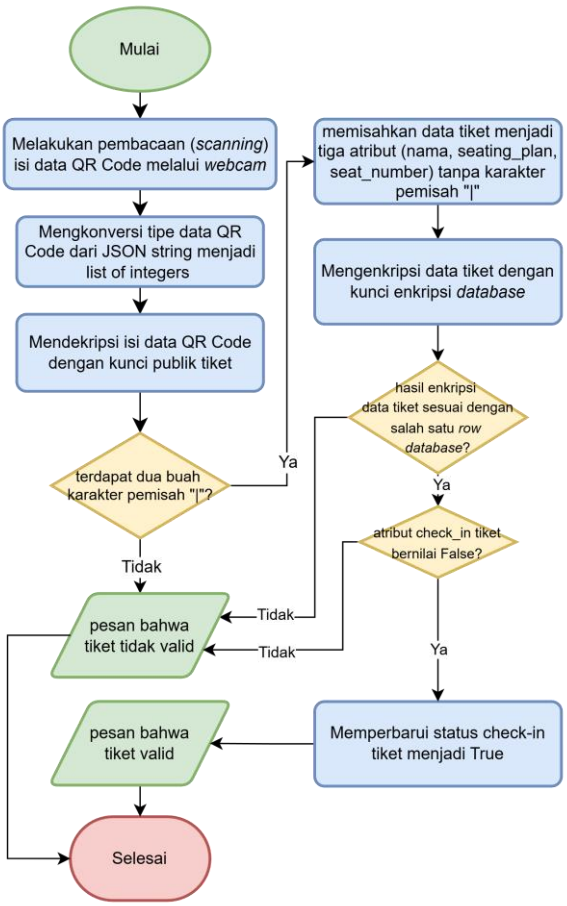
Data Output | Explain | Messages | Notifications

	id [PK] bigint	name text	email text	phone text	seating_plan text	seat_number text	check_in boolean
69	70	[4068536657...	[4758922089324233...	[42724002950917...	[968416261210819...	[1208404548295...	false
70	71	[4068536657...	[4758922089324233...	[42724002950917...	[968416261210819...	[1208404548295...	false
71	73	[2987295113...	[4758922089324233...	[42724002950917...	[410961155280463...	[2024202192430...	true
72	72	[4068536657...	[4758922089324233...	[42724002950917...	[274762573906292...	[1208404548295...	true
73	75	[1093994053...	[4758922089324233...	[42724002950917...	[427104648490621...	[1208404548295...	true
74	74	[4288325544...	[4758922089324233...	[42724002950917...	[410961155280463...	[5505605357616...	true
75	76	[3598125989...	[4758922089324233...	[42724002950917...	[427104648490621...	[1783923033547...	true
76	77	[4694853757...	[4758922089324233...	[42724002950917...	[410961155280463...	[1783923033547...	false
77	78	[2601629203...	[4758922089324233...	[42724002950917...	[410961155280463...	[4721717639219...	true
78	79	[3341752689...	[4758922089324233...	[42724002950917...	[410961155280463...	[4389794949619...	true
79	80	[8355069089...	[4758922089324233...	[42724002950917...	[410961155280463...	[1783923033547...	true
80	81	[3647366525...	[4758922089324233...	[42724002950917...	[410961155280463...	[2873403685553...	true

Gambar 13. Hasil Penyimpanan Tiket Terenkripsi di *Database*

3) Implementasi Verifikasi Tiket

Alur verifikasi tiket dimulai dari pembacaan isi data tiket dalam QR Code melalui *scanning* hingga pencocokan data dengan *row database* dijelaskan melalui diagram alir pada gambar 14.



Gambar 14. Alur Verifikasi Tiket

Pengecekan keaslian tiket pada proses verifikasi dilakukan dengan tiga tahapan, yakni melakukan pengecekan terhadap dua karakter pemisah “|”, melakukan pengecekan hasil enkripsi

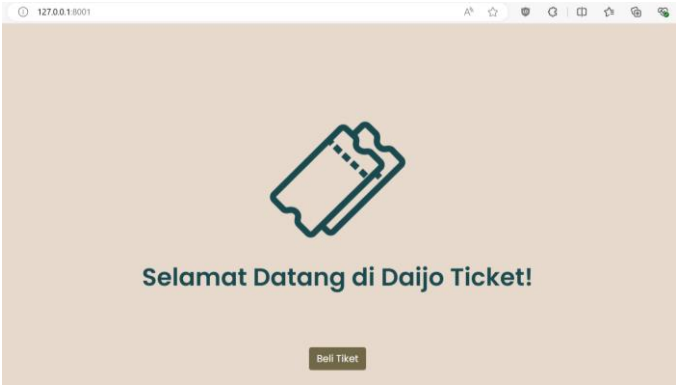
dengan *database*, dan melakukan pengecekan atribut check-in pada tiket. Contoh hasil verifikasi tiket ditunjukan pada subbab selanjutnya yakni implementasi antarmuka.

4) Implementasi Antarmuka

Implementasi antarmuka pengguna diwujudkan dalam bentuk *website customer* dan *website* admin. Implementasi antarmuka pengguna dibuat dengan *framework* Django, HTML, CSS dan Bootstrap. *Framework* Django sebagai *backend* berperan dalam menangani logika aplikasi dan interaksi *database*, sementara HTML, CSS, dan Bootstrap berperan sebagai *frontend* untuk memastikan halaman *website* memiliki struktur, desain, dan responsivitas yang baik.

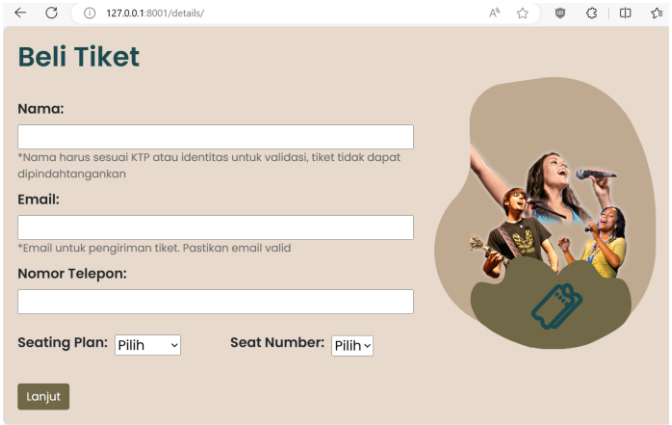
a) Implementasi Antarmuka Website Customer

Antarmuka *website customer* terdiri dari empat halaman, yakni halaman *home*, halaman pembelian tiket, halaman konfirmasi pembelian, dan halaman terima kasih. Halaman *home* adalah halaman awal dari *website customer*. Halaman ini berisi logo Daijo Tiket, pesan selamat datang, serta tombol "Beli Tiket" yang akan mengarahkan pengguna ke halaman pembelian tiket.



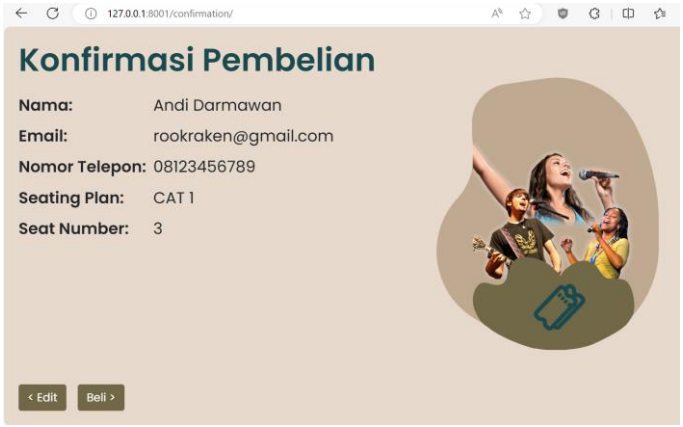
Gambar 15. Antarmuka *Website Customer*: Halaman *Home*

Pada halaman pembelian tiket, pengguna dapat melakukan pembelian tiket dengan mengisi informasi pembelian berupa nama, email, nomor telepon, pilihan tempat duduk (*seating plan*) serta nomor tempat duduk (*seat number*). Terdapat tombol lanjut yang akan menyimpan sementara data pembelian dan membawa pengguna ke halaman konfirmasi pembelian.



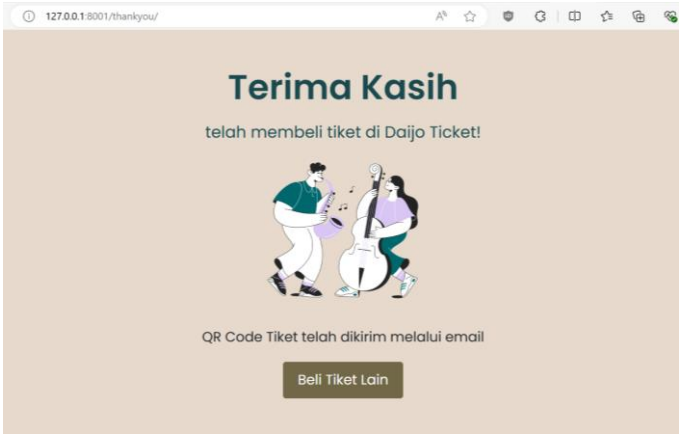
Gambar 16. Antarmuka *Website Customer*: Halaman Pembelian Tiket

Selanjutnya, pengguna dapat melihat data pembelian sementara miliknya pada halaman konfirmasi pembelian. Pengguna diberikan opsi untuk melakukan pengeditan informasi dengan menekan tombol "Edit" yang terletak pada kiri bawah halaman. Apabila pengguna yakin informasi pembelian sudah benar, pengguna dapat melakukan *checkout* dengan menekan tombol "Beli" yang terletak di bagian kiri bawah halaman di samping tombol "Edit".



Gambar 17. Antarmuka *Website Customer*: Halaman Konfirmasi Pembelian

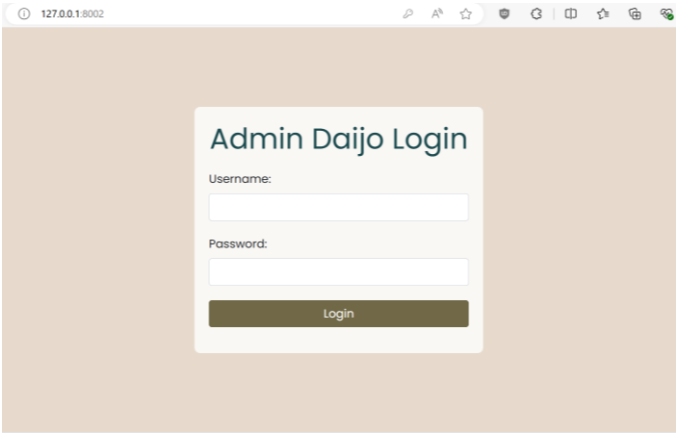
Terakhir, halaman terima kasih akan muncul usai pengguna melakukan *checkout* pembelian dengan menekan tombol "Beli" pada halaman "konfirmasi pembelian". Pada halaman ini terdapat pesan terima kasih atas pembelian tiket serta informasi bahwa tiket dalam bentuk QR Code telah dikirim melalui email. Pengguna juga diberikan opsi untuk membeli tiket lagi dengan menekan tombol "Beli Tiket Lain" yang akan mengarahkan pengguna kembali ke halaman *home*.



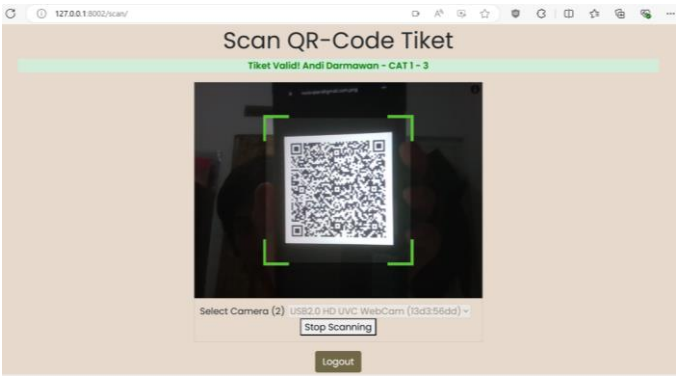
Gambar 18. Antarmuka *Website Customer*: Halaman Terima Kasih

b) Implementasi Antarmuka Website Admin

Antarmuka *website* admin terdiri halaman *login* dan halaman *scan* tiket. Pada halaman *login*, terdapat *field username* dan *field password* yang harus diisi dengan benar oleh pengguna untuk dapat melakukan *login* dan berpindah ke halaman *scan* tiket. Pada halaman *scan* tiket, pengguna dapat memindai tiket dengan terlebih dahulu memilih kamera/*webcam*. Terdapat pula tombol "Logout" untuk pengguna mengakhiri sesi dan kembali ke halaman *login*.

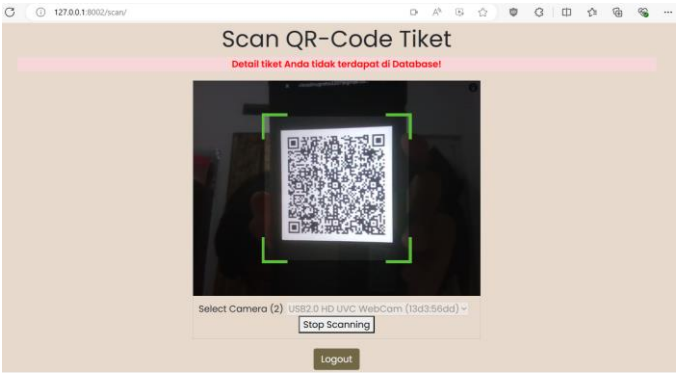


Gambar 19. Antarmuka *Website Admin*: Halaman Login



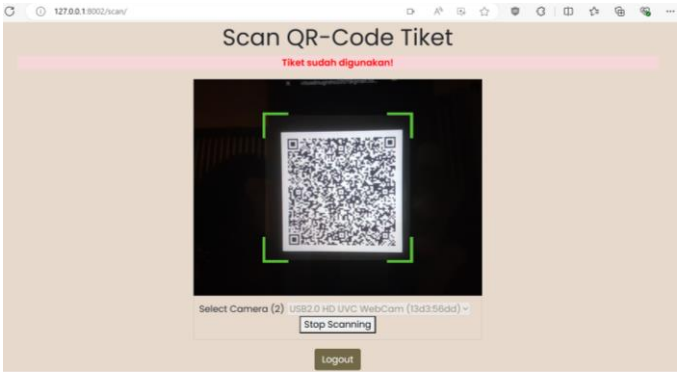
Gambar 20. Antarmuka *Website Admin*: Halaman Scan Tiket saat Mendeteksi Tiket Valid

Gambar 20 merupakan tampilan halaman *scan* tiket pada *website* admin usai mendeteksi tiket yang valid sesuai dengan ketentuan proses verifikasi tiket. Pada kasus tiket valid, terdapat pesan berwarna hijau yang bertuliskan "Tiket Valid" beserta detail pembelian tiket dengan format "Nama – *Seating Plan* – *Seat Number*."



Gambar 21. Antarmuka *Website Admin*: Halaman Scan Tiket saat Mendeteksi Tiket yang Tidak Valid

Gambar 21 merupakan tampilan halaman *scan* tiket pada *website* admin usai mendeteksi tiket yang tidak valid sesuai dengan ketentuan proses verifikasi tiket. Pada kasus tiket tidak valid, terdapat pesan berwarna merah yang menyatakan bahwa detail tiket tidak terdapat di *database* (tidak valid).



Gambar 22. Antarmuka *Website Admin*: Halaman Scan Tiket saat Mendeteksi Tiket yang Sudah Pernah Digunakan (Tidak Valid)

Gambar 22 merupakan tampilan halaman *scan* tiket pada *website* admin usai mendeteksi tiket yang tidak valid karena terdeteksi sudah pernah dilakukan *check-in* sesuai dengan ketentuan proses verifikasi tiket. Pada kasus tiket tidak valid karena sudah pernah *check-in*, terdapat pesan berwarna merah yang menyatakan bahwa detail tiket sudah digunakan (tidak valid).

V. PENGUJIAN

Setelah melakukan implementasi, sistem *e-ticketing* yang telah dikembangkan diuji dengan tujuan memastikan tiap komponen sistem bekerja sesuai dengan spesifikasi sehingga mampu memenuhi ekspektasi sistem: 1. meningkatkan keamanan penggunaan *e-ticket* (memastikan keaslian tiket dan mencegah pembacaan data tiket), 2. menjamin kepemilikan *e-ticket*, dan 3. mempertahankan kemudahan proses validasi *e-ticket* (proses verifikasi tiket yang cepat). Pengujian dilakukan dengan *functional testing* (menguji ekspektasi 1 dan 2) serta *non-functional testing* (menguji ekspektasi 3).

A. *Functional Testing*

Functional testing dilakukan menggunakan 14 skenario pengujian dengan metode *blackbox testing* dan *whitebox testing*. Pengujian dengan metode *blackbox* mengambil sudut pandang *customer* dan admin dalam menggunakan sistem, dengan menggunakan berbagai skenario positif dan skenario negatif. *Blackbox testing* dilakukan untuk menjawab tujuan pengujian yakni memastikan tiap komponen sistem bekerja sesuai dengan spesifikasi sehingga mampu memenuhi ekspektasi sistem. steganografi dan mempermudah proses verifikasi tiket.

Pengujian dengan metode *whitebox* dilakukan menggunakan fungsi logger pada python untuk menampilkan kondisi (*state*) program proses per proses pada *command prompt*, memastikan proses-proses sistem dilakukan secara benar dan tepat. *Whitebox testing* dilakukan untuk memastikan ekspektasi sistem nomor 1 dan 2 terpenuhi.

Tabel 4. Rincian Hasil *Functional Testing*

Status	Definisi / Isi Kolom	Jumlah
PASS	Skenario uji berhasil dijalankan dengan hasil yang diharapkan tanpa	14

	adanya *error* atau *bug* tertentu	
FAIL	Skenario uji gagal memberikan hasil yang diharapkan karena adanya *error* atau *bug* tertentu	0
UNEXECUTED	Skenario uji tidak dijalankan karena adanya ketergantungan sistem, masalah *environment*, atau alasan lainnya	0
TOTAL EXECUTED		14

Berdasarkan rincian hasil pengujian *functional testing* pada tabel 4, disimpulkan bahwa sistem telah berfungsi secara **baik** dan **sesuai** dengan ekspektasi pengujian maupun dengan spesifikasi yang telah didefinisikan.

B. Non-Functional Testing

Non-functional testing bertujuan untuk mengukur aspek kualitas sistem diluar aspek fungsional yang ada, yakni untuk mengukur kecepatan proses verifikasi tiket dengan berbagai kunci yang diestimasikan ideal jika kurang dari 0.5 detik/tiket. Pengujian ini dilakukan untuk memberikan pengalaman pengguna yang optimal, yakni mendapatkan proses verifikasi tiket yang cepat sehingga berdampak positif terhadap antrian pengunjung di *venue event*.

Tabel 5. Rincian Hasil *Non-Functional Testing*

No	Panjang Kunci	Panjang Cipherteks	Percobaan ke-	Waktu Verifikasi Tiket (detik)	Rata-Rata (detik)
1	512-760 bit	200-250 karakter	1	0.013	0.0134
			2	0.017	
			3	0.012	
			4	0.014	
			5	0.011	
2	1024-1530 bit	400-500 karakter	1	0.059	0.0612
			2	0.085	
			3	0.064	
			4	0.045	
			5	0.053	
3	2048-3036 bit	900-1000 karakter	1	0.906	1.0596
			2	0.752	
			3	1.411	
			4	0.631	
			5	1.598	

Berdasarkan hasil pengujian non-fungsional pada tabel 5, disimpulkan bahwa panjang kunci yang optimal bagi sistem untuk dapat melakukan verifikasi tiket (dengan waktu kurang dari 0.5 detik) adalah kunci dengan panjang **1024-1530 bit** yang diperoleh dengan membangkitkan bilangan prima p dan q sepanjang 512 bit. Dengan panjang kunci ini, didapatkan waktu verifikasi tiket sebesar 0.06 detik. Waktu ini adalah waktu tambahan di luar waktu fisik atau waktu yang dibutuhkan *customer* untuk menunjukan tiket ke mesin pemindai dan waktu responsivitas mesin pemindai terhadap QR Code. Perkiraan waktu verifikasi tambahan pada berbagai *event* dengan simulasi 100 dan 500 customer per *gate event* (diasumsikan terdapat banyak *gate event* untuk jumlah customer yang lebih besar) adalah 1 menit dan 5 menit waktu tambahan yang dinilai cukup kecil dan *applicable*.

VI. KESIMPULAN DAN SARAN

Berdasarkan penelitian yang telah dilakukan, didapat kesimpulan sebagai berikut:

1. Sistem *e-ticketing* berbasis media pindai untuk *event* pertunjukan berhasil dikembangkan dengan menerapkan konsep *digital signature* menggunakan algoritma Schmidt-Samoa dan media pindai QR Code. Sistem ini dapat meningkatkan keamanan penggunaan tiket dengan memastikan keaslian tiket dan mencegah pembacaan data tiket di *database*, sekaligus menjamin kepemilikan tiket.
2. Dengan penggunaan kunci tiket dan kunci *database* sepanjang 1024 bit, rata-rata waktu verifikasi tambahan yang dibutuhkan sistem untuk menentukan keaslian tiket adalah 0.06 detik. Kecepatan verifikasi ini berdampak positif dalam mempercepat penanganan antrian pengunjung di lokasi *event* pertunjukan yang ramai.

Adapun saran untuk pengembangan yang didapatkan dari pengerjaan tugas akhir ini adalah sebagai berikut:

1. Mengembangkan sistem yang mempunyai aspek non-fungsional yang lebih baik dari aspek *availability* (ketersediaan *website* dengan waktu yang lama) dan *performance* (mampu menangani pemrosesan paralel).
2. Mengembangkan sistem *super-admin* yang memiliki wewenang atas pengelolaan seluruh admin dan dapat melakukan re-generasi kunci untuk *event* baru yang secara otomatis menggantikan kunci lama.
3. Mengintegrasikan sistem dengan komponen-komponen perangkat keras misalnya perangkat QR scanner yang dapat mempercepat proses pemindaian tiket.
4. Mengembangkan sistem yang tidak memerlukan konektivitas internet untuk proses validasi tiket dengan *database.*
5. Mengembangkan sistem yang mendukung proses pembayaran tiket.

ACKNOWLEDGMENT

Terima kasih saya ucapkan kepada Tuhan Yang Maha Esa karena atas kasih dan berkat-Nya yang berlimpah penulis dapat menyelesaikan makalah tugas akhir yang berjudul "Pengembangan Sistem E-Ticketing Berbasis QR Code untuk Event Pertunjukan dengan Implementasi Algoritma Schmidt-Samoa" dengan tepat waktu. Terima kasih saya ucapkan juga kepada Bapak Dr. Ir. Rinaldi Munir, M.T selaku pembimbing tugas akhir yang telah membantu saya selama penyusunan tugas akhir ini. Tak lupa juga terima kasih saya ucapkan kepada para penguji, orang tua penulis, Ibu Dr. Fetty Fitriyanti Lubis, S.T., M.T. selaku dosen wali, seluruh Bapak dan Ibu dosen ITB, teman-teman pyxis nautica, teman-teman jayaplaza squad, teman-teman anggota MBWG ITB, serta seluruh teman-teman ITB yang tidak dapat penulis sebutkan satu persatu. Semoga hasil makalah ini dapat menjadi inspirasi bagi seluruh orang yang membaca untuk dapat menerapkan serta mengembangkan keseluruhan bahasan makalah ini.

REFERENSI

[1] Eka, Abidin. 2021. "Proses Pelayanan Sistem E-ticketing Pada Kmp. Agung Samudra Ix Di Pelabuhan Ketapang – Gilimanuk Oleh Pt. Pelayaran Agung Samudera Di Dermaga Lcm." *Repository Universitas Maritim AMNI (UMINAR AMNI) Semarang*.

[2] E. Nurdiansyah and I. Afrianto. 2017. "IMPLEMENTASI QRCODE SEBAGAI TIKET MASUK EVENT DENGAN MEMPERHITUNGKAN TINGKAT KOREKSI KESALAHAN." *Jurnal Teknologi dan Informasi (JATI)*. vol 7, no 2.

[3] Bennet, Joanne. 2020. "Journal of Data Protection & Privacy." *Henry Stewart Publications*. vol 4. no 1 winter 2020-2021: 93-99.

[4] Irwan, C. 2011. "Enkripsi Pada QR Code Tiket dengan RSA." *Informatika STEI ITB*.

[5] Vinton, K. 2014. "Seven Arrested For Fleecing Stubhub For $1.6 Million In Tickets." *Forbes*. Diakses dari https://www.forbes.com/sites/katevinton/2014/07/23/seven-arrested-for-fleecing-stubhub-for-1-6-million-in-tickets/ pada tanggal 14 Agustus 2024.

[6] Suprihadi. 2023. "Penyebab dan Dampak Kerusuhan Konser Coldplay di Indonesia." *Kompasiana*. Diakses dari https://www.kompasiana.com/nexiumtsamarah1130/6558488396b6804b383344b3/penyebab-dan-dampak-kerusuhan-konser-coldplay-di-indonesia pada tanggal 14 Agustus 2024.

[7] Hellman, Martin E. 1978. "An overview of public key cryptography." *IEEE Communications Magazine 40*, no 5: 42-49.

[8] Fajar, Revi M. 2006. "Kriptografi Kunci Publik." Makalah Matematika Diskrit Program Studi Teknik Informatika, Institut Teknologi Bandung.

[9] R.L. Rivest, A. Shamir, dan L. Adleman. 1978. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Commun. ACM*: 120-126.

[10] Schmidt-Samoa K. 2006. "A New Rabin-type Trapdoor Permutation Equivalent to Factoring. Electronic Notes in Theoretical Computer Science." *Department of Computer Science, Darmstadt University of Technology*: 79-94.

[11] Al-Haija, Qasem Abu, Mohamad M. Asad, dan Ibrahim Marouf. 2018. "A Systematic Expository Review of Schmidt-Samoa Cryptosystem." *International Journal of Mathematical Sciences and Computing (IJMSC) 4*: 2-21. DOI: 10.5815/ijmsc.2018.02.02.

[12] Katz, Jonathan dan Yehuda Lindel. 2007. "Chapter 12: Digital Signature Schemes." *Introduction to Modern Cryptography*: 399-402.

[13] Soon, Tan Jin. 2008. "QR code." *synthesis journal 2008*:59-78.

[14] Ristanto, Willy, Willy Sudiarto Raharjo, dan Antonius Rachmat Chrismanto. 2013. "Implementasi Algoritma Schmidt-Samoa Pada Enkripsi Dekripsi Email Berbasis Android." *Jurnal Informatika Universitas Kristen Duta Wacana.* no 1.