Implementasi Tanda Tangan Digital pada Penjulan Tiket Konser

Gde Anantha Priharsena - 13519026
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13519026@std.stei.itb.ac.id

***Abstrak*—Penjualan tiket konser merupakan salah satu komponen pendukung pada suatu konser untuk. Penjualan tiket konser sering dilakukan secara daring karena dinilai lebih efektif. Hal ini dikarenakan calon penonton dapat membeli tiket dari mana saja dan kapan saja. Akan tetapi, hal ini justru membuat tiket konser terjual dengan sangat cepat. Sehingga, calon penonton memilih untuk membeli tiket konser pada para calo. Sehingga meningkatkan resiko pembelian tiket konser palsu. Oleh karena itu, dibutuhkan sebuah sistem penjualan dan verifikasi tiket untuk mencegah hal tersebut. Pada makalah ini akan dibahas pembuatan API yang merupakan sebuah sistem penjualan dan verifikasi tiket konser yang mengimplementasikan tanda tangan digital dengan algoritma RSA dan fungsi hash Keccak (SHA3).**

Kata Kunci—tiket konser; tanda tangan digital; RSA; keccak; API;

I. PENDAHULUAN

Sebelum masa Pandemi, berbagai konser sering diadakan di Indonesia. Konser merupakan sebuah pertunjukkan musik langsung yang dilakukan didepan banyak penonton[1]. Sebuah konser dapat dilakukan oleh seorang musisi atau ansabel musik seperti orkestra, paduan suara, band, dan lain-lain. Sebuah konser dapat dilakukan di berbagai tempat baik di tempat yang tertutup seperti klub malam, amfiteater, dan *concert hall* maupun tempat yang terbuka seperti taman, pantai, bahkan stadion yang memiliki kapasitas penonton yang tentunya sangat besar. Adapun beberapa konser yang cukup populer di Indonesia antara lain Hammersonic, Java Jazz Festival, Soundandrenaline, WeTheFest, Djakarta Warehouse Project (DWP), dan La La La Fest.

Suatu konser membutuhkan beberapa komponen untuk menyukseskan konser tersebut. Komponen-komponen tersebut antara lain tata letak musisi, tata letak peralatan musik, tata letak pencahayaan, hingga penonton termasuk penjualan tiket. Penjualan tiket dilakukan dengan berbagai cara untuk mendapatkan penonton sebanyak-banyaknya. Akan tetapi, cara penjualan tiket yang paling populer adalah penjualan tiket secara daring, biasanya menggunakan situs web. Karena cara ini dianggap cara yang paling mudah dan efektif dalam proses penjualan tiket. Dengan penjualan tiket secara daring, tiket konser dapat dibayarkan dengan metode transfer, kartu kredit, atau metode pembayaran-pembayaran daring lainnya[2]. Sehingga, penonton tidak perlu menghabiskan waktunya untuk datang ke lokasi penjualan tiket dan antre berjam-jam hanya untuk membeli tiket konser tersebut. Penonton dapat langsung membeli tiket konser yang dijual oleh panitia konser dari mana saja dan kapan saja dengan kemudahan berbagai metode pembayaran yang disediakan.

Akan tetapi, dibalik kemudahan pembelian tiket konser secara daring terdapat beberapa oknum-oknum yang memanfaatkan hal tersebut untuk melakukan penipuan dalam penjualan tiket konser. Salah satu kasus penipuan penjualan tiket konser di Indonesia adalah saat penjualan tiket konser grup band One Direction yang dilakukan pada tahun 2015 lalu[3]. Menurut CNBC Indonesia, terdapat 12% orang yang mengalami penipuan saat membeli tiket konser[4]. Hal ini disebabkan banyaknya praktik pembelian tiket yang tidak wajar karena menggunakan *bot-bot* yang dapat membeli tiket dalam jumlah banyak dengan waktu yang sangat cepat saat penjulan tiket dimulai. Selanjutnya, tiket-tiket yang dibeli oleh *bot* tersebut akan dijual kembali dengan harga yang jauh lebih mahal dari harga yang ditawarkan oleh panitia konser sebenarnya. Sehingga, orang-orang akan memilih untuk menggunakan jasa calo yang meningkatkan angka penipuan dari penjualan tiket konser tersebut. Konsekuensinya adalah banyaknya tiket konser palsu yang dijual oleh para calo.

Untuk mencegah calon penonton konser membeli tiket konser palsu yang dijual oleh para calo adalah membuat sebuah sistem penjualan tiket yang dilengkapi fitur verifikasi tiket yang dapat dilakukan oleh para calon penonton yang hendak membeli sebuah tiket konser. Verifikasi tiket dilakukan dengan menyesuaikan tanda tangan digital yang dicantumkan pada tiket dengan atribut-atribut lainnya yang ada pada tiket seperti id tiket, tipe tiket, harga tiket, nomor kursi, dan harga. Sehingga dengan proses verifikasi tiket ini para calon penonton dapat membuktikan keaslian dari tiket yang mereka beli dan tidak perlu khawatir untuk ditolak masuk ke lokasi konser karena membeli tiket palsu.

II. LANDASAN TEORI

A. Tanda Tangan Digital

Tanda tangan digital adalah sebuah mekanisme untuk melakukan verifikasi terhadap keaslian sebuah pesan. Tanda tangan digital adalah nilai kriptografis yang bergantung pada isi pesan dan kunci[5]. Tidak seperti tanda tangan dokumen cetak pada umumnya, tanda tangan digital selalu berbeda-beda antara satu isi dokumen dengan dokumen lainnya. Terdapat beberapa tipe implementasi dari algoritma tanda tangan digital namun umumnya beroperasi pada dasar yang sama yaitu menggunakan

konsep matematika dan fungsi hash satu arah untuk mendeteksi apakah isi sebuah pesan telah mengalami manipulasi atau masih merupakan pesan asli. Terdapat tiga fungsi utama dalam melakukan verifikasi tanda tangan digital.

1. Authentication

Tanda tangan digital memberikan keyakinan pada penerima pesan bahwa pesan benar diubah oleh pihak yang berotoritas. Saat melakukan verifikasi tanda tangan digital, pesan diverifikasi dengan menggunakan kunci publik yang disediakan oleh pengirim pesan. Apabila pesan berhasil diverifikasi dengan menggunakan kunci publik tersebut, maka penerima pesan dapat yakin bahwa pesan yang diterima benar-benar dibuat oleh pengirim yang mempunyai kunci privat. Hal ini disebabkan kunci publik dan kunci privat dari sebuah mekanisme tanda tangan digital bersifat pasangan dan tidak akan bekerja apabila nilai dari kunci publik atau kunci privat nya diubah.

2. Data Integrity

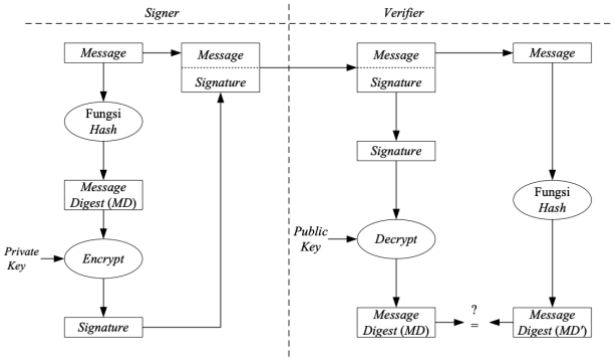
Mekanisme tanda tangan digital juga memungkinkan penerima pesan untuk melakukan pemeriksaan terhadap keaslian dari isi pesan. Hal ini penting dilakukan karena isi pesan yang dikirimkan melalui medium tertentu beresiko mengalami rekayasa sehingga isi pesan yang diterima sudah berbeda dengan pesan asli yang dikirimkan. Dalam domain tertentu seperti dunia perbankan, hal ini menjadi sangat krusial.

3. Non-Repudiation

Non-repudiation dalam mekanisme verifikasi tanda tangan digital artinya seseorang tidak dapat melakukan penyangkalan terhadap pesan yang pernah dikirimkan. Hal ini berbasis pada poin pertama yaitu sebuah pesan hanya dapat dikirimkan oleh pengirim yang memiliki kunci privat, oleh sebab itu identitas pengirim dapat dipastikan dan pengirim tidak dapat menyangkal hal tersebut.

Terdapat dua cara mendatangani pesan secara digital yaitu dengan mengenkripsi pesan serta menggunakan kombinasi fungsi hash (*hash function*) dan kriptografi kunci-publik. Penandatanganan digital dengan cara mengenkripsi pesan dapat dilakukan dengan menggunakan kriptografi simetri dan menggunakan kriptografi kunci publik. Dengan menggunakan kriptografi simetri, pesan yang dienkripsi dengan algoritma simetri sudah memberikan solusi untuk otentikasi pengirim karena kunci simetri hanya diketahui oleh pengirim dan penerima. Akan tetapi, cara ini tidak menyediakan mekanisme untuk anti-penyangkalan. Dengan menggunakan kriptografi kunci publik, pesan dienkripsi dengan kunci privat pengirim dan pesan didekripsi dengan kunci publik pengirim sehingga kerahasiaan pesan dan otentikasi keduanya dicapai sekaligus. Sementara, penandatangan pesan secara digital dengan pendekatan menggunakan kombinasi fungsi hash (*hash function*) dan kriptografi kunci-publik dapat dilakukan untuk beberapa kasus yang hanya membutuhkan otentikasi tetapi tidak dengan kerahasiaan pesan dapat. Adapun langkah-langkah pemberian tanda tangan dengan pendekatan menggunakan kombinasi fungsi hash (*hash function*) dan kriptografi kunci-publik sebagai berikut.

1. Pengirim menghitung nilai hash dari pesan yang ingin dikirim.
2. Pengirim mengenkripsi nilai hash dengan kunci privatnya menggunakan persamaan enkripsi sesuai dengan kriptografi kunci-publik yang digunakan dan menghasilkan tanda tangan digital dari pesan tersebut.
3. Pengirim mentransmisikan pesan yang diikuti dengan tanda tangan digitalnya ke penerima.
4. Penerima menghitung nilai hash dari pesan yang diterima.
5. Penerima melakukan dekripsi terhadap tanda tangan digital yang disertakan pada pesan yang dikirimkan pengirim dengan kunci publik si pengirim dengan persamaan dekripsi sesuai dengan kriptografi kunci-publik yang digunakan.
6. Kemudian, penerima membandingkan nilai hash dari pesan yang diterima dengan teks yang dihasilkan dari dekripsi terhadap tanda tangan digital tersebut. Jika keduanya memiliki nilai yang sama maka pesan dan tanda tangan digital tersebut otentik (berhasil diverifikasi). Jika nilainya berbeda, maka dapat dipastikan tanda tangan tersebut tidak otentik sehingga pesan dianggap tidak asli.



Gambar 1. Proses Pemberian dan Verifikasi Tanda Tangan Digital

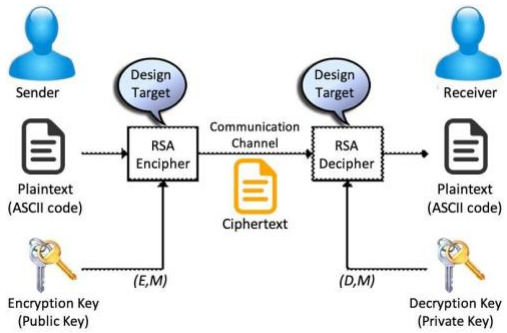
Sumber: Munir, Rinaldi. "Tanda Tangan Digital ". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Tanda-tangan-digital-2020.pdf. Diakses pada 20 Desember 2021.

B. Algoritma RSA

Algoritma RSA merupakan algoritma enkripsi kunci publik[6]. Dengan demikian, algoritma RSA menggunakan kunci yang berbeda saat melakukan proses enkripsi dan dekripsi. Terdapat pasangan kunci publik dan kunci privat pada algoritma RSA. Sesuai dengan namanya, kunci publik adalah bagian kunci yang bebas tersedia kepada publik sedangkan kunci privat adalah bagian kunci yang hanya diketahui oleh pengirim pesan. Apabila kunci privat diketahui oleh pihak eksternal, maka keseluruhan skema algoritma RSA sudah tidak aman. Skema kerja algoritma RSA dalam pengiriman pesan oleh Alice kepada Bob adalah sebagai berikut:

1. Alice memilih dua buah angka prima yang besar yaitu p dan q.

2. Alice menghitung n = p * q.
3. Alice memilih sebuah angka e sebagai kunci publik yang bukan merupakan faktor dari (p-1) * (q-1). Nilai e diberikan oleh Alice kepada Bob.
4. Alice memilih sebuah angka d sebagai kunci privat yang memenuhi persamaan (d*e) mod (p1)*(q-1)=1.
5. Alice menghitung message digest dari pesan yang akan dikirimkan, *message digest* ini lalu dienkripsi dengan kunci privat yang tidak dibagikan kepada siapapun.
6. Bob menerima pesan yang dikirimkan oleh Alice lalu melakukan dekripsi *message digest* dengan kunci publik yang diberikan oleh Alice.
7. Setelah melakukan dekripsi terhadap *message digest*, Bob menghitung *message digest* berdasarkan isi pesan yang diterimanya. Apabila kedua *message digest* sama, maka pesan telah terverifikasi.
8. Apabila *message digest* yang diterima Bob berbeda dengan hasil perhitungannya maka pesan yang diterima Bob sudah mengalami rekayasa.



Gambar 2. Algoritma RSA

Sumber: researchgate.net

C. Fungsi Hash Satu Arah

Fungsi *hash* satu arah adalah fungsi *hash* yang bekerja dalam satu arah. Pesan yang diubah menjadi *message digest* tidak dapat diubah kembali kedalam pesan semula. Sifat-sifat dari fungsi *hash* satu arah antara lain,

1. Fungsi hash (F) dapat diterapkan pada blok data ukuran apa saja.
2. F menghasilkan nilai (d) dengan panjang tetap.
3. F(x) mudah dihitung untuk setiap nilai x yang diberikan.
4. Untuk setiap d yang dihasilkan, tidak mungkin dikembalikan nilai x sedemikian sehingga F(x) = d. itulah sebabnya fungsi F dikatakan fungsi hash satu arah (one-way hash function).
5. Untuk setiap x yang diberikan, tidak mungkin mencari y ≠ x sedemikian sehingga F(y) = F(x).
6. Tidak mungkin mencari pasangan x dan y sedemikian sehingga F(x) = F(y).

D. Algoritma Keccak (SHA3)

Keccak merupakan algoritma fungsi hash satu arah yang berbasis pada konstruksi spon dengan menggunakan fungsi permutasi keccak-f dengan rentang (panjang) permutasi b, ukuran setiap lane dimana: b = 25 x 2l dengan $0 \leq l \leq 6$ [7]. Algoritma keccak memiliki prinsip yang sama dengan algoritma cipher block, dimana proses dilakukan terhadap blok-blok, setiap hasil proses bergantung dari masukan dan hasil proses sebelumnya, serta setap proses dikenakan pada sebuah fungsi utama yang terdiri dari sejumlah round fungsi yang diiterasi beberapa kali. Namun terdapat perbedaan antara algoritma hash satu arah Keccak, dengan algoritma cipher block, yaitu:

1. Keccak tidak memiliki key-schedule.
2. Menggunakan konstanta round yang bersifat tetap dari pada round key.

Keccak menggunakan inner state selama proses hashing berlangsung. Fungsi spon yang digunakan terdiri dari padding, absorbing, dan squeezing. Setiap state memiliki panjang sesuai dengan panjang permutasi, yaitu b. Algoritma keccak menerima tiga parameter masukan, yaitu *bitrate* (r), *capacity* (c), dan *diversity* (d). Secara umum proses dari keccak ini adalah:

1. Preproses pesan masukan (P), yaitu menerapkan *padding* pada pesan masukan. Panjang pesan masukan hasil *padding* harus merupakan kelipatan r, dengan r = *bitrate*.
2. Pemecahan pesan masukan menjadi P0, P1, P2, … , Pi , dimana i = jumlah kelipatan panjang *bitrate* untuk panjang pesan masukan.
3. *Absorbing* pada semua pecahan pesan masukan.
4. *Squeezing* sebanyak j, dimana j = kelipatan panjang keluaran r/w untuk memenuhi panjang keluaran yang diinginkan, r = *bitrate* dan w = panjang *lane* dari *state*. dengan: w = 2l.
5. Keluaran merupakan konkatenasi dari keluaran *squeezing* pada rentang *bitrate* tertentu.

III. RANCANGAN SOLUSI

Tiket merupakan sebuah kertas kecil yang memberikan hak akses kepada pemegangnya untuk memasuki sebuah lokasi atau mengikuti sebuah acara. Tiket konser digunakan untuk memverifikasi seseorang untuk menghadiri suatu acara konser. Tiket konser dapat dibeli sebelum konser tersebut dimulai. Akan tetapi, terdapat beberapa oknum yang memalsukan tiket yang mereka jual. Solusi yang akan diimplementasikan untuk mencegah calon penonton konser membeli tiket konser palsu yang dijual oleh para calo adalah membuat sebuah sistem penjualan tiket yang dilengkapi fitur verifikasi tiket yang dapat dilakukan oleh para calon penonton yang hendak membeli sebuah tiket konser.

Proses verifikasi tiket ini dapat memanfaatkan tanda tangan digital yang dicantumkan pada tiket konser yang berkaitan. Proses pemberian tanda tangan digital dilakukan dengan

menggunakan kombinasi antara fungsi hash dengan algoritma Keccak (SHA3) dan kriptografi kunci-publik RSA. Selanjutnya penonton dapat melakukan verifikasi tiket yang dia beli pada situs web yang disediakan oleh panitia konser dengan memasukkan beberapa atribut yang diperlukan id tiket, tipe tiket, nomor kursi tiket, harga tiket, dan tanda tangan digital yang dicantumkan pada tiket tersebut. Sehingga dengan proses verifikasi tiket ini para calon penonton dapat membuktikan keaslian dari tiket yang mereka beli dan tidak perlu khawatir untuk ditolak masuk ke lokasi konser karena membeli tiket konser yang palsu.

IV. Implementasi Solusi

A. Implementasi Tanda Tangan Digital

Implementasi tanda tangan digital dilakukan dengan menggunakan kombinasi antara fungsi hash dengan algoritma Keccak (SHA3) dan kriptografi kunci-publik RSA. Untuk mengimplementasikan hal tersebut dibutuhkan 3 kelas, yaitu Keccak, RSA, dan Signature.

1. Kelas Keccak

   Kelas Keccak diimplementasikan untuk melakukan *hashing* pada tiket dengan algoritma Keccak mulai dari *padding*, *squeezing*, hingga *absorbing*. Hal tersebut akan menghasilkan *message digest* dari tiket yang akan dibeli oleh calon penonton.

2. Kelas RSA

   Kelas RSA diimplementasikan untuk membangkitkan kunci privat dan kunci publik dari setiap tiket, mengenkripsi nilai *message digest* dari hasil *hashing* untuk menghasilkan tanda tangan digital untuk setiap tiket, dan mendekripsi tanda tangan digital dari tiket yang dibeli oleh penonton.

3. Kelas Signature

   Kelas Signature diimplementasikan untuk memberikan tanda tangan digital dari setiap tiket yang akan dibeli oleh calon penonton dan memverifikasi tanda tangan digital dari tiket yang sudah dibeli oleh penonton.

B. Implementasi Sistem Penjualan dan Verifikasi Tiket

Sistem penjualan dan verifikasi tiket dilakukan dengan tipe antarmuka Application Programming Interface (API) dengan tujuh *endpoint*.

TABEL I. DAFTAR ENDPOINT API

No.	Endpoint	Method	Deskripsi
1.	/ticket	GET	*Endpoint* ini digunakan untuk menampilkan detail dari seluruh tiket konser yang tersedia pada basis data tiket.
2.	/ticket/{id}	GET	*Endpoint* ini digunakan untuk menampilkan detail tiket konser dengan id tertentu yang tersedia pada basis data tiket.
3.	/ticket	POST	*Endpoint* ini digunakan untuk menambahkan sebuah tiket berikut dengan kunci privat dan kunci publiknya ke basis data tiket.
4.	/ticket/{id}	PUT	*Endpoint* ini digunakan untuk mengubah detail tiket konse dengan ID tertentu pada basis data tiket.
5.	/ticket/{id}	DELETE	*Endpoint* ini digunakan untuk menghapus tiket konser dengan ID tertentu dari basis data tiket.
6.	/buy/{id}	PUT	*Endpoint* ini digunakan untuk mekanisme pembelian tiket konser dengan ID tertentu pada basis data dengan mengubah atribut available dari tiket tersebut menjadi False.
7.	/verify	POST	*Endpoint* ini digunakan untuk memverifikasi tiket konser yang dibeli oleh penonton berdasarkan tanda tangan digital dan atribut-atribut tiket yang tercantum pada tiket tersebut.

1. Penampilan Detail Seluruh Tiket Konser

Pada *endpoint* /ticket dengan *method* GET, panitia konser dapat melihat detail dari seluruh tiket konser yang tersedia pada basis data tiket. Berikut merupakan contoh *response* dari *endpoint* ini.

Endpoint	/ticket
Method	GET
Response	[{ "id": 1, "type": "Platinum", "seat": "A1", "price": 1000000, "available": true, "public_key1": 309059119, "public_key2": 1151038123, "private_key1": 818296399, "private_key2": 1151038123 },

	{
	"id": 2,
	"type": "Platinum",
	"seat": "A2",
	"price": 1000000,
	"available": false,
	"public_key1": 1633162337,
	"public_key2": 621216983,
	"private_key1": 97523297,
	"private_key2": 621216983
	}
	]

2. Penampilan Detail Tiket Konser dengan ID Tertentu

Pada *endpoint* /ticket/{id} dengan *method* GET, panitia konser dapat melihat detail dari tiket konser dengan ID tertentu yang tersedia pada basis data tiket. Berikut merupakan contoh *response* dari *endpoint* ini.

Endpoint	/ticket/1
Method	GET

Response	[
	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	"available": true,
	"public_key1": 309059119,
	"public_key2": 1151038123,
	"private_key1": 818296399,
	"private_key2": 1151038123
	}
	]

3. Penambahan Tiket Konser

Pada *endpoint* /ticket dengan *method* POST, panitia konser dapat menambahkan sebuah tiket konser ke basis data tiket. Pada penambahan tiket konser ini dilakukan pemanggilan terhadap method RSA.generateKey() untuk membangkitkan kunci publik dan kunci privat dari tiket konser tersebut. Berikut merupakan contoh *request payload* dan *response* dari *endpoint* ini.

Endpoint	/ticket
Method	POST

Request Payload	{
	"id": 3,
	"type": "Gold",
	"seat": "G1",
	"price": "500000",
	"available": true
	}

Response	[
	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	"available": true,
	"public_key1": 309059119,
	"public_key2": 1151038123,
	"private_key1": 818296399,
	"private_key2": 1151038123
	},
	{
	"id": 2,
	"type": "Platinum",
	"seat": "A2",
	"price": 1000000,
	"available": false,
	"public_key1": 1633162337,
	"public_key2": 621216983,
	"private_key1": 97523297,
	"private_key2": 621216983
	},
	{
	"id": 3,
	"type": "Gold",
	"seat": "G1",
	"price": 500000,
	"available": true,
	"public_key1": 1593269767,
	"public_key2": 711009569,
	"private_key1": 38027815,
	"private_key2": 711009569
	}
	]

4. Pengubahan Detail Tiket Konser dengan ID Tertentu

Pada *endpoint* /ticket/{id} dengan *method* PUT, panitia konser dapat mengubah detail dari tiket konser dengan ID tertentu yang tersedia pada basis data tiket. Pada pengubahan tiket konser ini dilakukan pemanggilan ulang terhadap method RSA.generateKey() untuk membangkitkan kunci publik dan

kunci privat dari tiket konser yang diubah detailnya. Berikut merupakan contoh *request payload* dan *response* dari *endpoint* ini.

Endpoint	/ticket/3
Method	PUT

Request Payload	{
	"id": 3,
	"type": "Gold",
	"seat": "G2",
	"price": "500000",
	"available": true
	}

Response	[{ "id": 1, "type": "Platinum", "seat": "A1", "price": 1000000, "available": true, "public_key1": 309059119, "public_key2": 1151038123, "private_key1": 818296399, "private_key2": 1151038123 }, { "id": 2, "type": "Platinum", "seat": "A2", "price": 1000000, "available": false, "public_key1": 1633162337, "public_key2": 621216983, "private_key1": 97523297, "private_key2": 621216983 }, { "id": 3, "type": "Gold", "seat": "G2", "price": 500000, "available": false, "public_key1": 1731786367, "public_key2": 7670483, "private_key1": 7455103, "private_key2": 7670483
	}]

5. Penghapusan Tiket Konser dengan ID Tertentu

Pada *endpoint* /ticket/{id} dengan *method* DELETE, panitia konser dapat menghapus tiket konser dengan ID tertentu yang tersedia pada basis data tiket. Berikut merupakan contoh dan *response* dari *endpoint* ini.

Endpoint	/ticket/3
Method	DELETE

Response	[
	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	"available": true,
	"public_key1": 309059119,
	"public_key2": 1151038123,
	"private_key1": 818296399,
	"private_key2": 1151038123
	},
	{
	"id": 2,
	"type": "Platinum",
	"seat": "A2",
	"price": 1000000,
	"available": false,
	"public_key1": 1633162337,
	"public_key2": 621216983,
	"private_key1": 97523297,
	"private_key2": 621216983
	}
	]

6. Pembelian Tiket Konser

Pada *endpoint* /buy/{id} dengan *method* PUT, penonton dapat membeli tiket konser dengan ID tertentu yang masih tersedia pada basis data tiket. Pada pembelian tiket konser juga dilakukan pemberian tanda tangan digital dengan kunci publik dan kunci privat tiket tersebut. Pemberian tanda tangan digital memanggil fungsi Signature.sign() yang meliputi pemanggilan fungsi Keccak.hash() dan RSA.encrypt() yang menambahkan atribut signature ke tiket yang dibeli oleh penonton tersebut. Berikut merupakan contoh *response* dari *endpoint* ini.

Endpoint	/buy/1

Method	PUT

Response	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	“signature”:
	"0304860128089648490810072000540497656944089648
	490803048601280800349692025103515110072000054111
	12651801111265180049765694402795402720103717471
	08003496920292689603012682355308313127390310144
	88011112651800292689603100720005408003496920038
	96002803101448800304860128027954027211112651800
	29268960308964849080251035151003896002810072000
	54003896002801037174710831312739025103515108003
	49692051203847603101448800038960028100720005401
	26823553083131273903048601280103717471083131273
	90831312739111126518002795402720831312739111126
	51800512038476029268960310072000540831312739051
	20384760072909539080034969208313127390512038476
	027954027202926896030038960028"
	}

7. Verifikasi Tiket Konser

Pada *endpoint* /verify dengan *method* POST, penonton dapat memverifikasi tiket konser yang mereka beli. Verifikasi tanda tangan digital memanggil fungsi Signature.verifySignedTicket() yang meliputi pemanggilan fungsi Keccak.hash() dan RSA.decrypt(). Hasil dari verifikasi tiket ini akan dijadikan sebuah atribut pada *response* dari *endpoint* ini yaitu is_ticket_valid yang bernilai *true* jika tiket tersebut asli dan *false* jika tiket tersebut palsu. Berikut merupakan contoh *request payload* dan *response* dari *endpoint* ini.

Endpoint	/verify
Method	POST

Request Payload	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	“signature”:
	"0304860128089648490810072000540497656944089648
	490803048601280800349692025103515110072000054111
	12651801111265180049765694402795402720103717471
	08003496920292689603012682355308313127390310144
	88011112651800292689603100720005408003496920038
	96002803101448800304860128027954027211112651800
	29268960308964849080251035151003896002810072000
	54003896002801037174710831312739025103515108003
	49692051203847603101448800038960028100720005401
	26823553083131273903048601280103717471083131273
	90831312739111126518002795402720831312739111126
	51800512038476029268960310072000540831312739051
	20384760072909539080034969208313127390512038476
	027954027202926896030038960028"
	}

Response	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	“signature”:
	"0304860128089648490810072000540497656944089648
	490803048601280800349692025103515110072000054111
	12651801111265180049765694402795402720103717471
	08003496920292689603012682355308313127390310144
	88011112651800292689603100720005408003496920038
	96002803101448800304860128027954027211112651800
	29268960308964849080251035151003896002810072000
	54003896002801037174710831312739025103515108003
	49692051203847603101448800038960028100720005401
	26823553083131273903048601280103717471083131273
	90831312739111126518002795402720831312739111126
	51800512038476029268960310072000540831312739051
	20384760072909539080034969208313127390512038476
	027954027202926896030038960028",
	"is_ticket_valid": true
	}

V. EKSPERIMEN DAN HASIL ANALISIS

Terdapat dua eksperimen yang dapat dilakukan dalam implementasi tanda tangan digital pada penjualan tiket konser ini yaitu dengan mengubah tanda tangan digital yang ada pada tiket tersebut dan mengubah atribut tiket yang ada pada tiket tersebut. Eksperimen ini dilakukan dengan adanya asumsi bahwa calo hanya bisa mengubah kedua hal tersebut dan tidak bisa mengubah kunci privat maupun kunci publik yang dimiliki oleh tiket tersebut karena hanya panitia konser yang dapat melakukan hal tersebut pada endpoint /ticket/id dengan *method* PUT.

A. Perubahan Tanda Tangan Digital

Berikut merupakan hasil eksperimen dengan mengubah tanda tangan digital dari tiket konser yang dibeli dengan ID = 1.

Endpoint	/verify
Method	POST

Request Payload	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	“signature”:
	"5555560128089648490810072000540497656944089648
	49080304860128080034969202510351511007200054111
	12651801111265180049765694402795402720103717471
	08003496920292689603012682355308313127390310144
	88011112651800292689603100720005408003496920038
	96002803101448800304860128027954027211112651800
	29268960308964849080251035151003896002810072000
	54003896002801037174710831312739025103515108003
	49692051203847603101448800038960028100720005401
	26823553083131273903048601280103717471083131273
	90831312739111126518002795402720831312739111126
	51800512038476029268960310072000540831312739051
	20384760072909539080034969208313127390512038476
	027954027202926896030038960028"
	}

Response	{
	"id": 1,
	"type": "Platinum",
	"seat": "A1",
	"price": 1000000,
	“signature”:
	"5555560128089648490810072000540497656944089648
	49080304860128080034969202510351511007200054111
	12651801111265180049765694402795402720103717471
	08003496920292689603012682355308313127390310144
	88011112651800292689603100720005408003496920038
	96002803101448800304860128027954027211112651800
	29268960308964849080251035151003896002810072000
	54003896002801037174710831312739025103515108003
	49692051203847603101448800038960028100720005401
	26823553083131273903048601280103717471083131273
	90831312739111126518002795402720831312739111126
	51800512038476029268960310072000540831312739051
	20384760072909539080034969208313127390512038476
	027954027202926896030038960028",
	"is_ticket_valid": false
	}

B. Perubahan Atribut Tiket

Berikut merupakan hasil eksperimen dengan mengubah salah satu atribut dari tiket yang dibeli dengan ID=1 yaitu nomor kursi dari “A1” menjadi “A2”.

Endpoint	/verify
Method	POST

Request Payload	{
	"id": 1,
	"type": "Platinum",
	"seat": "A3",
	"price": 1000000,
	“signature”:
	"0304860128089648490810072000540497656944089648
	49080304860128080034969202510351511007200054111
	12651801111265180049765694402795402720103717471
	08003496920292689603012682355308313127390310144
	88011112651800292689603100720005408003496920038
	96002803101448800304860128027954027211112651800
	29268960308964849080251035151003896002810072000
	54003896002801037174710831312739025103515108003
	49692051203847603101448800038960028100720005401
	26823553083131273903048601280103717471083131273
	90831312739111126518002795402720831312739111126
	51800512038476029268960310072000540831312739051
	20384760072909539080034969208313127390512038476
	027954027202926896030038960028"
	}

Response	{ "id": 1, "type": "Platinum", "seat": "A3", "price": 1000000, “signature”: "0304860128089648490810072000540497656944089648 49080304860128080034969202510351511007200054111 12651801111265180049765694402795402720103717471 08003496920292689603012682355308313127390310144 88011112651800292689603100720005408003496920038 96002803101448800304860128027954027211112651800 29268960308964849080251035151003896002810072000 54003896002801037174710831312739025103515108003 49692051203847603101448800038960028100720005401 26823553083131273903048601280103717471083131273 90831312739111126518002795402720831312739111126 51800512038476029268960310072000540831312739051 20384760072909539080034969208313127390512038476

	027954027202926896030038960028", "is_ticket_valid": false }

C. Hasil Analisis

Kedua hasil eksperimen gagal memverifikasi tiket yang dibeli oleh penonton tersebut. Hal ini dikarenakan kedua hasil eksperimen menghasilkan nilai yang berbeda antara hasil *hashing* dari tiket tersebut dengan teks yang dihasilkan dari dekripsi dari tanda tangan digital yang dicantumkan pada tiket tersebut. Berbeda dengan proses verifikasi tiket konser yang dilakukan pada subbab IV.7, dimana kondisi tersebut berhasil memverifikasi tiket konser yang dibeli oleh penonton karena menghasilkan nilai yang sama antara hasil *hashing* dari tiket tersebut dengan teks yang dihasilkan dari dekripsi dari tanda tangan digital yang dicantumkan pada tiket tersebut.

VI. Kesimpulan dan Saran

Tanda Tangan Digital dapat diimplementasikan dengan menggunakan kombinasi antara fungsi hash dengan algoritma Keccak (SHA3) dan kriptografi kunci-publik RSA. Tanda tangan digital dilakukan untuk memastikan tiket konser yang dibeli oleh penonton merupakan tiket yang asli sehingga menghindarkan calon penonton membeli tiket palsu yang dijual oleh para oknum yang tidak bertanggung jawab. Berdasarkan pengujian yang telah dilakukan pada bagian sebelumnya, dapat disimpulkan program yang diimplementasikan sukses untuk memverifikasi tiket konser baik yang asli maupun yang palsu.

Akan tetapi, masih banyak batasan asumsi yang digunakan pada pembuatan program ini antara lain program yang dibuat sebatas dengan tipe antarmuka *Application Program Interface* (API) dan belum memperhitungkan keamanan terkait hak akses dari setiap *endpoint* sehingga memungkinkan adanya pengguna-pengguna mengakses *endpoint* yang bukan hak nya seperti calon penonton yang mencoba mengakses endpoint /ticket. Sehingga diperlukan pengembangan lebih lanjut untuk membuat sistem penjualan dan verifikasi tiket yang dapat menjamin pembelian tiket yang lebih aman.

Pranala Kode Program

Kode program dapat diakses melalui pranala berikut https://github.com/gdeananthap/digital-signature-for-concert-ticket

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa karena atas berkat dan rahmat-Nya yang melimpah, penulis dapat menyelesaikan makalah ini. Penulis juga ingin mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, MT., selaku dosen mata kuliah IF4020 Kriptografi yang telah menyediakan bahan ajar melalui situs web sehingga memudahkan penulis dalam memahami mata kuliah ini. Penulis juga mengucapkan terima kasih kepada kedua orang tua dan keluarga penulis yang senantiasa mendoakan dan mendukung studi penulis.

Referensi

[1] Concert. Wikipedia. https://en.wikipedia.org/wiki/Concert. Diakses pada 20 Desember 2021.

[2] Kartini, K., Fahnun, B. U., & Pratiwi, D. (2013). Perancangan Sistem Informasi Pemesanan Tiket Konser Musik Online Berbasis Lokasi. *SEMNASTEKNOMEDIA ONLINE*, *1*(1), 27-25. Diakses pada 20 Desember 2021.

[3] Banyak Penipuan, Hati-Hati Beli Tiket Online Konser One Direction. detikHot. https://hot.detik.com/music/d-2868602/banyak-penipuan-hati-hati-beli-tiket-online-konser-one-direction. Diakses pada 20 Desember 2021.

[4] Hati-Hati! 12% Orang yang Beli Tiket Konser Online Kena Tipu. CNBC Indonesia. https://www.cnbcindonesia.com/lifestyle/20180915195502-33-33305/hati-hati-12-orang-yang-beli-tiket-konser-online-kena-tipu. Diakses pada 20 Desember 2021.

[5] Munir, Rinaldi. "Tanda Tangan Digital ". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Tanda-tangan-digital-2020.pdf. Diakses pada 20 Desember 2021.

[6] Munir, Rinaldi. "Algoritma RSA". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Algoritma-RSA-2020.pdf. Diakses pada 20 Desember 2021.

[7] Munir, Rinaldi. "SHA-3 (Keccak)". https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/SHA-3-2020. Diakses pada 20 Desember 2021.

PERNYATAAN

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

Bogor, 20 Desember 2021

Gde Anantha Priharsena / 13519026