Implementasi Algoritma Kunci Publik RSA, Fungsi *Hash* Keccak, dan Steganografi untuk Memberikan Tanda Tangan Digital pada Hasil Tes COVID-19

Nadya Laurentia - 18219071 (*Author*)
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: nadyalaurentia@gmail.com

***Abstract*—Pada masa pandemi ini, beberapa layanan masyarakat hanya dapat diakses jika memiliki bukti hasil tes COVID-19. Karena hasil tes tidak selalu tercatat pada akun PeduliLindungi, masyarakat seringkali hanya diminta menunjukkan bukti hasil tes COVID-19. Sayangnya, banyak kasus pemalsuan hasil tes COVID-19 demi kemudahan mendapatkan layanan. Oleh karena itu, diusulkan sebuah solusi untuk menjaga keaslian dan integritas hasil tes, yaitu implementasi tanda tangan digital pada hasil tes COVID-19 dengan memanfaatkan steganografi. Tanda tangan digital dibangun dari fungsi *hash* Keccak dan algoritma RSA untuk disisipkan pada *file* gambar hasil tes COVID-19 menggunakan metode LSB. Implementasi pembangkitan kunci, penandatanganan, serta verifikasi tanda tangan akan memanfaatkan *Representational State Transfer Application Programming Interface* (REST API). Implementasi solusi ini terbukti berhasil dalam melakukan validasi keaslian dan integritas hasil tes COVID-19. Ke depannya, solusi ini dapat distandardisasi dan diseragamkan agar bisa diterapkan di Indonesia secara luas.**

Keywords—tanda tangan digital; fungsi hash Keccak, kriptografi kunci publik RSA; metode LSB; tes COVID-19

I. PENDAHULUAN

Pandemi COVID-19 yang melanda dunia sejak akhir 2019 mengubah banyak aktivitas dalam kehidupan sehari-hari. Masyarakat dihimbau untuk berada di rumah jika tidak ada kepentingan atau kewajiban yang mengharuskan untuk berkegiatan di ruang publik. Di Indonesia, untuk dapat melakukan beberapa aktivitas di ruang publik, masyarakat diwajibkan melakukan tes COVID-19 dan menunjukkan buktinya kepada petugas yang bersangkutan. Hal ini bertujuan untuk memitigasi risiko terjadinya penyebaran COVID-19 pada suatu ruang publik. Hasil tes COVID-19 yang dilakukan pada instansi resmi seharusnya akan tercatat secara otomatis pada akun PeduliLindungi. Sayangnya, hasil tes, terutama tes antigen, terkadang tidak tercatat pada akun PeduliLindungi. Dengan demikian, masyarakat harus menunjukkan sendiri dokumen hasil tes yang didapat dari instansi kesehatan kepada petugas yang bersangkutan.

Salah satu aktivitas yang membutuhkan hasil tes COVID-19 adalah bepergian menggunakan transportasi umum seperti kereta api atau pesawat. Aturan ini sempat dicabut ketika keadaan pandemi di Indonesia mulai membaik pada bulan Maret 2022. Namun, peraturan ini kembali diberlakukan pada bulan April 2022. Ketika peraturan ini diberlakukan, terdapat banyak kasus pemalsuan hasil tes COVID-19. Pemalsuan yang dilakukan mulai dari membuat hasil tes sendiri, melakukan modifikasi terhadap hasil tes yang sudah tidak berlaku, atau memodifikasi hasil tes yang resmi. Motif pemalsuan ini biasanya untuk menghemat uang atau karena sangat ingin melakukan perjalanan. Padahal, hal ini berbahaya karena memiliki potensi penularan virus kepada penumpang lainnya.

Oleh karena itu, diperlukan metode yang dapat memastikan keaslian dan integritas hasil tes COVID-19. Pada makalah ini, diusulkan implementasi tanda tangan digital pada hasil tes COVID-19 dengan memanfaatkan algoritma RSA, fungsi *hash* Keccak, serta steganografi. Dengan penyisipan tanda tangan digital, pemalsuan hasil tes COVID-19 dapat dengan mudah dideteksi. Harapannya, implementasi ini dapat membantu pencegahan penyebaran COVID-19.

II. DASAR TEORI

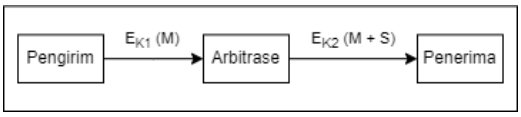
A. Tanda Tangan Digital

Tanda tangan digital adalah nilai kriptografis yang selalu berbeda antara satu dokumen dengan dokumen lainnya karena bergantung pada isi pesan dan kunci. Terdapat 2 cara untuk menghitung tanda tangan digital, yaitu sebagai berikut.

a) Mengenkripsi pesan

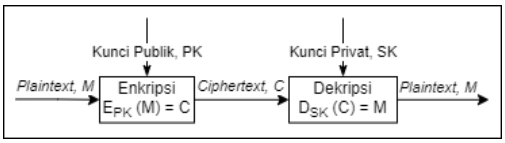
Enkripsi untuk tanda tangan digital dapat menggunakan kriptografi simetri ataupun kriptografi kunci publik. Jika menggunakan kriptografi simetri, dibutuhkan pihak arbitrase (penengah) yang dipercaya oleh kedua pihak, pengirim dan penerima. Pengirim dan penerima masing-masing akan memiliki kunci simetri rahasia dengan pihak arbitrase, misalnya K_1 untuk pengirim dan K_2 untuk penerima. Cara ini menjamin keaslian (autentikasi), kerahasiaan, serta menyediakan mekanisme anti-penyangkalan. Diagram alur

tanda tangan digital menggunakan enkripsi kriptografi simetri ditunjukkan pada Gambar 1, dengan E adalah enkripsi, M adalah pesan (*message*), dan S adalah tanda tangan (*signature*).



Gambar 1. Diagram alur tanda tangan digital menggunakan kriptografi simetri

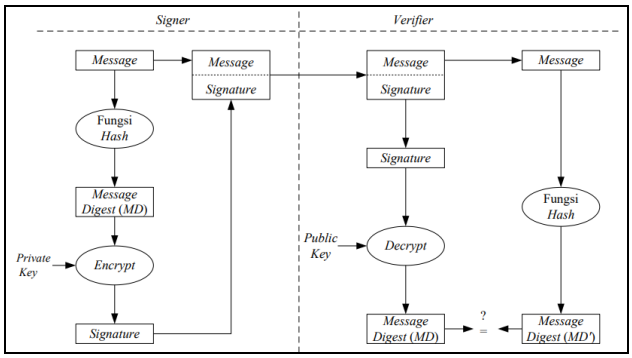
Jika menggunakan kriptografi kunci publik, pesan akan dienkripsi menggunakan kunci privat pengirim. Pesan tersebut kemudian akan didekripsi menggunakan kunci publik pengirim oleh penerima. Ide yang dicetuskan oleh Diffie dan Hellman ini juga menawarkan autentikasi, kerahasiaan, serta mekanisme anti-penyangkalan. Diagram alur tanda tangan digital menggunakan enkripsi kriptografi kunci publik ditunjukkan pada Gambar 2.



Gambar 2. Diagram alur tanda tangan digital menggunakan kriptografi kunci publik

b) Menggunakan kombinasi fungsi hash *dan kriptografi kunci publik*

Pada beberapa kasus, seringkali pesan tidak perlu dirahasiakan. Pesan hanya perlu dipastikan autentikasinya dan dilengkapi dengan mekanisme anti-penyangkalan. Kombinasi kriptografi kunci publik dan fungsi *hash* dapat digunakan untuk kasus tersebut. Tanda tangan digital dibuat dengan cara menghitung nilai *hash* dari pesan, kemudian mengenkripsinya dengan menggunakan kunci privat pengirim. Pengirim akan mengirimkan pesan beserta tanda tangan digital kepada penerima. Penerima akan mendekripsi tanda tangan digital kemudian membandingkan hasil *hash* tersebut dengan *hash* pesan yang dilakukan oleh penerima sendiri. Jika nilai *hash* sama, artinya pesan yang diterima sesuai dengan yang dikirimkan. Sebaliknya, jika nilai *hash* berbeda, pesan sudah mengalami perubahan. Diagram alur tanda tangan digital menggunakan enkripsi kriptografi kunci publik ditunjukkan pada Gambar 3.



Gambar 3. Diagram alur tanda tangan digital menggunakan kombinasi fungsi *hash* dan kriptografi kunci publik (Sumber: Slide Kuliah II4031 Kriptografi dan Koding: Tanda-tangan digital oleh Rinaldi Munir)

Pada makalah ini, akan digunakan kombinasi fungsi *hash* dan kriptografi kunci publik untuk membuat tanda tangan digital.

B. Algoritma Kunci Publik RSA

Ide untuk kriptografi kunci publik muncul pada tahun 1976 dalam makalah "*New Directions in Cryptography*" yang ditulis oleh Diffie dan Hellman. Algoritma untuk kriptografi kunci publik yang paling terkenal dan paling banyak digunakan adalah algoritma RSA yang diciptakan pada tahun 1976. RSA merupakan singkatan dari nama pencipta algoritma ini, yaitu Rivest, Shamir, dan Adleman, yang merupakan peneliti dari Massachussets Institute of Technology (MIT).

Persamaan enkripsi pada algoritma RSA ditunjukkan pada persamaan (1) sedangkan persamaan dekripsinya ditunjukkan pada persamaan (2).

$$E_e(m) = c = m^e \bmod n \quad (1)$$

$$D_d(c) = m = c^d \bmod n \quad (2)$$

Fungsi enkripsi E menggunakan kunci enkripsi e untuk mengubah pesan asli m menjadi pesan cipher c. Sementara itu, fungsi dekripsi D menggunakan kunci dekripsi d untuk mengembalikan pesan cipher c menjadi pesan asli m. Nilai e dan c tidak rahasia sedangkan nilai d dan m rahasia. Kemudian, terdapat juga n yang merupakan hasil perkalian dua buah bilangan prima rahasia yang nilainya besar. Nilai n sendiri tidak dirahasiakan. Keamanan algoritma RSA terletak pada sulitnya memfaktorkan n yang merupakan bilangan bulat yang besar menjadi faktor-faktor prima.

C. Fungsi Hash *Keccak*

Fungsi *hash* Keccak merupakan pemenang dalam kompetisi pengembangan fungsi *hash* baru yang bernama SHA-3. Kompetisi yang diselenggarakan National Institute of Standards and Technology (NIST) mengumumkan kompetisi pada tahun 2007 dan menentukan pemenang pada bulan Oktober tahun 2012. Keccak didesain oleh Breton, Daemen, Peeters, dan Van Assche.

Fungsi *hash* Keccak menggunakan *sponge construction*, yaitu memanfaatkan fungsi non-kompresi untuk *absorb*

(menyerap) kemudian *squeeze* (memeras) *digest*. Dengan mekanisme *absorb* dan *squeeze* ini, fungsi *hash* Keccak dapat mencegah *collision*.

D. Steganografi Digital

Steganografi adalah ilmu dan seni menyembunyikan pesan rahasia dengan suatu cara sedemikian rupa sehingga tidak menimbulkan kecurigaan akan keberadaan pesan tersebut. Kriteria steganografi yang baik adalah pesan rahasia tidak dapat dipersepsi secara visual atau audial (*imperceptiblе*), dokumen yang digunakan sebagai media penyembunyian tidak banyak berubah akibat penyisipan pesan rahasia (*fidelity*), pesan rahasia dapat diekstraksi kembali (*recovery*), dan ukuran yang disembunyikan sedapat mungkin besar (*capacity*).

Steganografi digital adalah suatu teknik menyembunyikan pesan digital di dalam suatu dokumen digital agar tidak terdeteksi keberadaannya. Dokumen digital yang digunakan sebagai media penyembunyian pesan (*cover object*) dapat berupa teks, audio, gambar, ataupun video. Pada makalah ini, jenis *cover object* yang digunakan adalah gambar. Metode steganografi yang akan digunakan adalah metode LSB (*Least Significant Byte*). Metode ini merupakan *spatial domain methods* yang memodifikasi langsung nilai *byte* dari *cover-object*. Pada metode LSB, bit LSB dari *pixel* gambar akan digantikan dengan bit pesan rahasia. Karena perubahan *byte* hanya satu lebih tinggi atau lebih rendah dari nilai sebelumnya, perubahan tidak akan berpengaruh terhadap persepsi visual. Ukuran pesan yang mampu disembunyikan oleh gambar berwarna (RGB) ditunjukkan pada persamaan (3) dengan S adalah ukuran pesan rahasia dalam satuan *byte* dan P adalah ukuran pixel *cover object*.

$$S = P \times P \times 3 / 8 \quad (3)$$

III. Rancangan Solusi dan Implementasi

Berikut adalah rancangan solusi untuk permasalahan yang telah dibahas pada bagian Pendahuluan.

A. Deskripsi Umum Solusi

Untuk memastikan keaslian dan integritas hasil tes COVID-19, akan disisipkan tanda tangan digital menggunakan steganografi pada *file* gambar hasil tes. Tanda tangan digital sendiri akan dibuat dengan implementasi *hash* Keccak yang kemudian dilindungi dengan kriptografi kunci publik RSA. Fungsi *hash* Keccak dipilih karena fungsi ini memiliki kemampuan untuk mencegah *collision*. Algoritma RSA sendiri digunakan karena telah terbukti keamanannya dan telah digunakan secara luas. Dengan penyisipan tanda tangan digital ini, hasil tes COVID-19 dapat dipastikan autentik (*authentication*), tidak diubah (*integrity*), dan tidak dapat disangkal (*non-repudiation*).

Solusi akan diimplementasikan oleh pihak instansi yang mengeluarkan hasil tes COVID-19. Hasil tes ini kemudian akan diverifikasi oleh pihak-pihak yang menyatakan hasil tes COVID-19 sebagai persyaratan untuk memberi layanan atau fasilitas, contohnya pihak maskapai penerbangan.

B. Rancangan Solusi

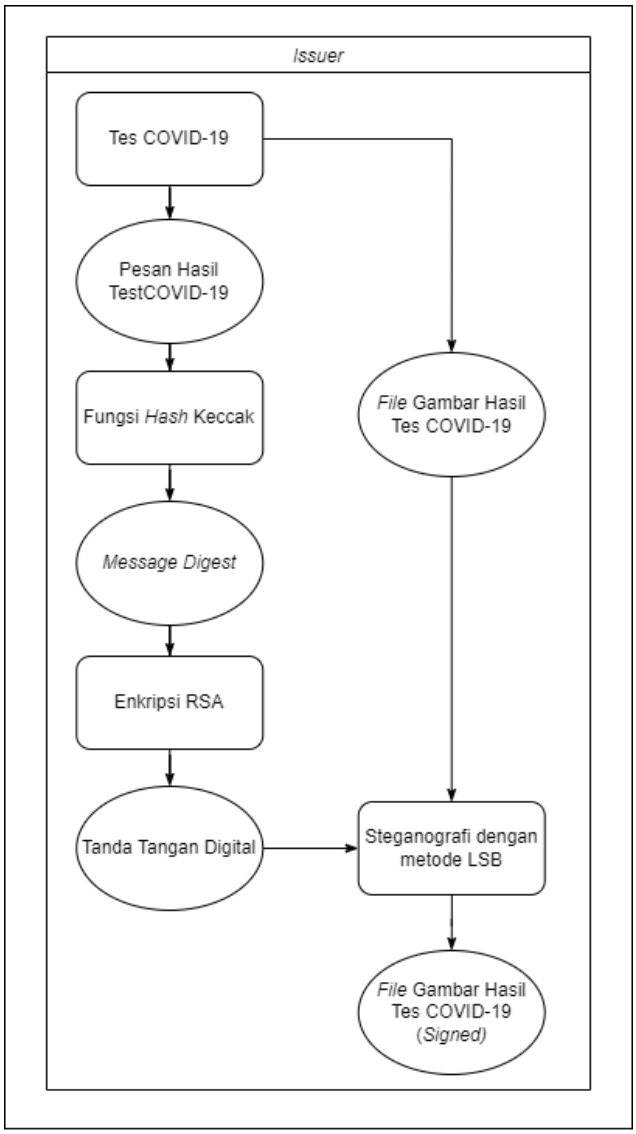
Secara garis besar, solusi yang dirancang dapat dibagi menjadi dua, yaitu rancangan untuk pihak instansi yang mengeluarkan hasil tes COVID-19, yang selanjutnya akan disebut sebagai *issuer*, dan rancangan untuk pihak yang memberikan persyaratan hasil tes COVID-19 untuk memberikan fasilitas, yang selanjutnya akan disebut sebagai *verificator*.

Solusi yang dirancang untuk pihak *issuer* merupakan proses penandatanganan digital dan proses steganografi digital. Langkah-langkah yang dilakukan oleh pihak *issuer* adalah sebagai berikut.

- *Issuer* membuat pasangan kunci publik dan kunci privat
- *Issuer* mengeluarkan hasil tes COVID-19 berbentuk gambar dan berbentuk pesan dengan format: "---START---<nama instansi>-<nama pelanggan>-<NIK pelanggan>-<tanggal lahir pelanggan>-<jenis kelamin pelanggan>-<nomor HP pelanggan>-<email pelanggan>-<jenis tes>-<hasil tes>---END---". Format tanggal adalah DD/MM/YYYY. Jenis kelamin pelanggan hanya memiliki 2 jenis *input*, yaitu "Laki-laki" atau "Perempuan". Hasil tes hanya memiliki juga 2 jenis *input*, yaitu "Positif" atau "Negatif".
- Pesan akan di-*hash* oleh *issuer* menggunakan fungsi *hash* Keccak.
- Hasil *message digest* dari *hash* dienkripsi oleh menggunakan kunci privat *issuer* untuk membentuk tanda tangan digital.
- *Issuer* menyisipkan tanda tangan digital ke dalam hasil tes COVID-19 yang berbentuk gambar menggunakan metode LSB.
- *Issuer* mengirimkan hasil tes COVID-19 yang telah disisipi tanda tangan digital kepada pelanggan.
- *Issuer* mempublikasikan kunci publik pada portal resmi instansi.

Berikut adalah arsitektur rancangan solusi untuk pihak *issuer*.

$ax + b = \chi$. (1)

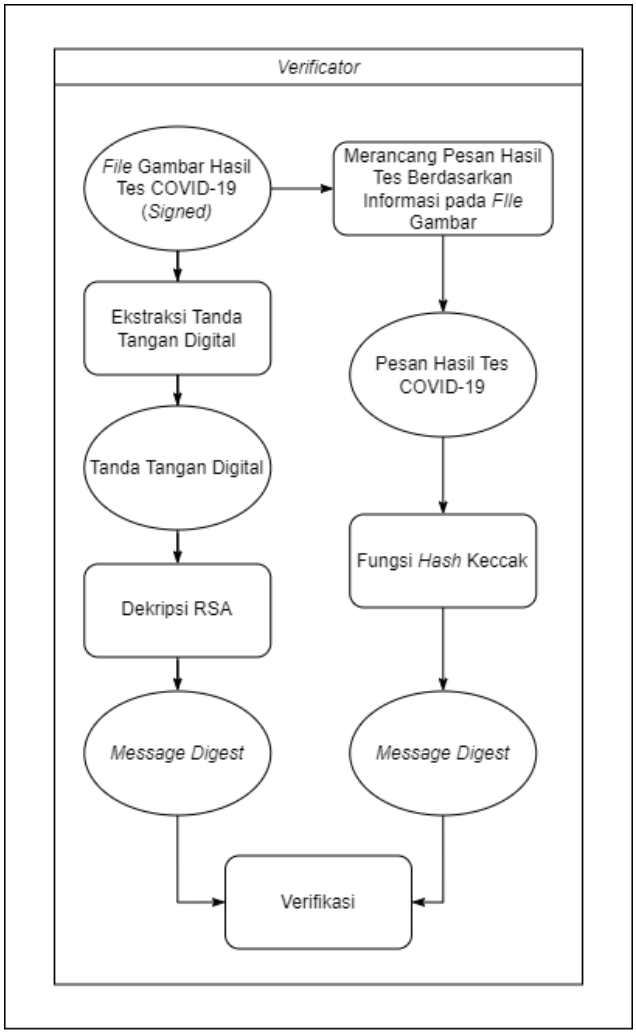


Gambar 4. Arsitektur Rancangan Solusi untuk Pihak *Issuer*

Solusi yang dirancang untuk pihak *verificator* merupakan proses mengekstrak tanda tangan digital yang disisipkan ke dalam hasil tes COVID-19 dan melakukan verifikasi tanda tangan tersebut. Langkah-langkah yang dilakukan oleh pihak *verificator* adalah sebagai berikut.

- *Verificator* menerima hasil tes COVID-19 dari pelanggan.
- *Verificator* melakukan ekstrasi tanda tangan digital dari gambar hasil tes COVID-19.
- *Verificator* mengakses kunci publik instansi.
- *Verificator* melakukan deskripsi tanda tangan digital sehingga mendapatkan hasil *hash* pesan.
- *Verificator* membuat pesan serupa dengan format yang telah didefinisikan sebelumnya sesuai dengan informasi yang tertera pada gambar hasil tes COVID-19.
- *Verificator* mengimplementasikan fungsi *hash* Keccak pada pesan yang dibuat.
- *Verificator* membandingkan hasil *hash* yang baru dibuat dengan hasil *hash* dari tanda tangan digital. Jika hasil *hash* sama, artinya hasil tes COVID-19 terverifikasi autentikasi dan integritasnya. Sebaliknya, jika hasil *hash*-nya berbeda, hasil tes COVID-19 sudah dimodifikasi.

Berikut adalah arsitektur rancangan solusi untuk pihak *verificator*.



Gambar 5. Arsitektur Rancangan Solusi untuk Pihak *Verificator*

C. Implementasi

Untuk implementasi solusi, akan digunakan *Representational State Transfer Application Programming Interface* (REST API). Pihak *issuer* akan membuat kunci publik dan kunci privat kemudian mempublikasikan kunci publik pada portal resmi instansi. Pihak *issuer* kemudian memberikan gambar hasil tes COVID-19 kepada pelanggan. Pelanggan kemudian akan menyerahkan gambar hasil tes COVID-19 kepada pihak *verificator*. Pihak *verificator* akan melakukan ekstraksi tanda tangan digital kemudian melakukan verifikasi menggunakan kunci publik *issuer* yang didapat dari portal resmi instansi yang bersangkutan.

Terdapat beberapa *endpoint* API yang akan dibangun, yaitu *endpoint* proses pembangkitan kunci, *endpoint* proses penandatanganan digital, dan *endpoint* proses verifikasi.

1) Proses Pembangkitan Kunci (`/generate`)

Endpoint untuk proses pembangkitan kunci menerima *request* HTTP dengan *method* POST dengan *request payload* berupa nama instansi. *Response payload* adalah pasangan kunci publik dan kunci privat RSA.

Request payload (JSON)

```
{
  "nama_institusi": "Pamame Farmasi
  Bandung"
}
```

Response payload (JSON)

```
{
  "kunci_publik":
  101613831794584083229494775550590830105798361613203321227471251334179234585903,
  "kunci_privat":
  326193312448716145912257706393572333682414570656779814103883238260038877897212771815530192002143191654515064164446248409445318634438785994747521464572308 7
}
```

2) Proses Penandatanganan Digital (`/sign`)

Endpoint untuk proses penandatanganan digital menerima *request* HTTP dengan *method* POST dengan *request payload* berupa informasi tes COVID-19. *Response payload* adalah gambar hasil tes COVID-19 yang sudah disisipi tanda tangan digital dengan metode LSB.

Request payload (JSON)

```
{
  "nama_instansi": "Pamame Farmasi
  Bandung",
  "nama_pelanggan": "Nadya Laurentia",
  "NIK_pelanggan": 123456789101112,
  "tanggal_lahir_pelanggan":
  "01/01/2001",
  "jenis_kelamin_pelanggan":
  "Perempuan",
  "nomor_HP_pelanggan": 081234567890,
  "email_pelanggan": "nadya@gmail.com",
  "jenis_tes": "COVID-19 Rapid Test
  Antigen",
  "tanggal_tes": "10/05/2022",
  "hasil_tes": "Negatif"
}
```

Response payload (JSON)

```
{
  "file_hasil_tes":
  "hasiltes_signed.jpg"
}
```

Gambar hasil tes akan diunggah ke *cloud storage* milik instansi. Pelanggan akan diberikan akses ke *cloud storage* menggunakan *email* terdaftar dan dapat mengunduh *file* gambar hasil tes.

3) Proses Verifikasi (`/verify`)

Endpoint untuk proses verifikasi hasil tes COVID-19 menerima *request* HTTP dengan *method* POST dengan *request payload* berupa informasi tes COVID-19, *file* hasil tes COVID-19, dan kunci publik instansi *issuer* yang bersangkutan. *Response payload* adalah hasil verifikasi yaitu "Verified" jika verifikasi valid atau "Modified" jika terdeteksi modifikasi pada *file* hasil tes COVID-19.

Request payload (JSON)

```
{
  "nama_instansi": "Pamame Farmasi
  Bandung",
  "nama_pelanggan": "Nadya Laurentia",
  "NIK_pelanggan": 123456789101112,
  "tanggal_lahir_pelanggan":
  "01/01/2001",
  "jenis_kelamin_pelanggan":
  "Perempuan",
  "nomor_HP_pelanggan": 081234567890,
  "email_pelanggan": "nadya@gmail.com",
  "jenis_tes": "COVID-19 Rapid Test
  Antigen",
  "tanggal_tes": "10/05/2022",
  "hasil_tes": "Negatif",
  "file_hasil_tes":
  "hasiltes_signed.jpg",
  "kunci_publik":
  101613831794584083229494775550590830105798361613203321227471251334179234585903
}
```

Response payload (JSON)

```
{
  "hasil_verifikasi": "Verified"
}
```

IV. Pengujian dan Pembahasan

A. Pengujian

Solusi yang telah diimplementasikan akan diuji untuk melihat pemenuhan target, yaitu untuk memastikan keaslian dan integritas hasil tes COVID-19. Berikut adalah kasus pengujian yang akan dilakukan.

TABLE I. KASUS PENGUJIAN

No	*Input*	Ekspektasi *Output*
1	Gambar hasil tes COVID-19 yang valid dan pesan hasil tes COVID-19 yang valid	`"hasil_verifikasi": "Verified"`
2	Gambar hasil tes COVID-19 yang valid dan pesan hasil tes COVID-19 yang tidak valid	`"hasil_verifikasi": "Modified"`
3	Gambar hasil tes COVID-19 yang tidak valid dan pesan hasil tes COVID-19 yang tidak valid	`"hasil_verifikasi": "Modified"`

Pengujian hanya akan dilakukan pada *endpoint* `/verify` menggunakan *input* gambar yang sudah ditandatangani menggunakan *endpoint* `/sign`. Tanda tangan digital yang digunakan untuk pengujian menggunakan kunci publik dan kunci privat yang telah dibangkitkan menggunakan *endpoint* `/generate` berikut.

```
{
  "kunci_publik":
  10161383179458408322949477555059083010
  57983616132033212274712513341792345859
  03,
  "kunci_privat":
  32619331244871614591225770639357233368
  24145706567798141038832382600388778972
  12771815530192002143191654515064164446
  24840944531863443878599474752146457230
  87
}
```

Berikut adalah penjelasan untuk setiap kasus pengujian.

1) Kasus Gambar dan Pesan Hasil Tes COVID-19 Valid

Pada kasus ini, *request payload* yang dikirimkan pada *endpoint* `/verify` adalah sebagai berikut.

```
{
  "nama_instansi": "Klinik Indeksa
  Bandung",
  "nama_pelanggan": "James Jo",
  "NIK_pelanggan": 123456789101112,
  "tanggal_lahir_pelanggan":
  "01/01/2001",
  "jenis_kelamin_pelanggan": "Laki-
  laki",
  "nomor_HP_pelanggan": 081234567890,
  "email_pelanggan": "jamesj@gmail.com",
  "jenis_tes": "COVID-19 Rapid Test
  Antigen",
  "tanggal_tes": "22/05/2022",
  "hasil_tes": "Positif",
  "file_hasil_tes":
  "hasiltes_signed.jpg",
  "kunci_publik":
  10161383179458408322949477555059083010
  57983616132033212274712513341792345859
  03
}
```

File hasiltes_signed.jpg yang dikirim pada *payload* di atas adalah sebagai berikut.

Gambar 6. Gambar Hasil Tes COVID-19 hasiltes_signed.jpg

Response payload yang diterima adalah sebagai berikut.

```
{
  "hasil_verifikasi": "Verified"
}
```

Response di atas menunjukkan hasil tes COVID-19 valid.

2) Kasus Gambar Hasil Tes COVID-19 Valid Namun Pesan Tidak Valid

Pada kasus ini, *request payload* yang dikirimkan pada *endpoint* `/verify` adalah sebagai berikut.

```
{
  "nama_instansi": "Klinik Indeksa
  Bandung",
  "nama_pelanggan": "James Jo",
  "NIK_pelanggan": 123456789101112,
  "tanggal_lahir_pelanggan":
```

```
  "01/01/2001",
  "jenis_kelamin_pelanggan":      "Laki-
  laki",
  "nomor_HP_pelanggan": 081234567890,
  "email_pelanggan": "jamesj@gmail.com",
  "jenis_tes":   "COVID-19   Rapid   Test
  Antigen",
  "tanggal_tes": "22/05/2022",
  "hasil_tes": "Negatif",
  "file_hasil_tes":
  "hasiltes_signed.jpg",
  "kunci_publik":
  10161383179458408322949477555059083010
  57983616132033212274712513341792345859
  03
}
```


File hasiltes_signed.jpg yang dikirim pada *payload* di atas adalah sebagai berikut.

Gambar 7. Gambar Hasil Tes COVID-19 hasiltes_signed.jpg

Response payload yang diterima adalah sebagai berikut.


```
{
  "hasil_verifikasi": "Modified"
}
```


Response di atas menunjukkan hasil tes COVID-19 tidak valid.

3) Kasus Gambar Hasil Tes COVID-19 dan Pesan Tidak Valid

Pada kasus ini, *request payload* yang dikirimkan pada *endpoint* `/verify` adalah sebagai berikut.


```
{
  "nama_instansi":    "Klinik    Indeksa
  Bandung",
  "nama_pelanggan": "James Jo",
  "NIK_pelanggan": 123456789101112,
  "tanggal_lahir_pelanggan":
  "01/01/2001",
  "jenis_kelamin_pelanggan":      "Laki-
  laki",
  "nomor_HP_pelanggan": 081234567890,
  "email_pelanggan": "jamesj@gmail.com",
  "jenis_tes":   "COVID-19   Rapid   Test
  Antigen",
  "tanggal_tes": "22/05/2022",
  "hasil_tes": "Negatif",
  "file_hasil_tes":
  "hasiltes_signed_modified.jpg",
  "kunci_publik":
  10161383179458408322949477555059083010
  57983616132033212274712513341792345859
  03
}
```


File hasiltes_signed_modified.jpg yang dikirim pada *payload* di atas adalah sebagai berikut.

Gambar 8. Gambar Hasil Tes COVID-19 hasiltes_signed_modified.jpg

Response payload yang diterima adalah sebagai berikut.

```
{
   "hasil_verifikasi": "Modified"
}
```

Response di atas menunjukkan hasil tes COVID-19 tidak valid.

B. Pembahasan

Berikut adalah pembahasan untuk pengujian yang telah dilakukan.

1) Kasus Gambar dan Pesan Hasil Tes COVID-19 Valid

Kasus pengujian ini melibatkan *input* yang valid. Berdasarkan hasil pengujian yang telah dilakukan di atas, *output response payload* yang dihasilkan sesuai dengan ekspektasi. Artinya, pengujian untuk kasus ini berhasil.

2) Kasus Gambar Hasil Tes COVID-19 Valid Namun Pesan Tidak Valid

Kasus pengujian ini melibatkan *input* gambar yang valid namun pesan tidak valid. Seharusnya, hasil tes positif, namun pesan dituliskan hasil tes negatif. Berdasarkan hasil pengujian yang telah dilakukan di atas, *output response payload* yang dihasilkan sesuai dengan ekspektasi. Artinya, pengujian untuk kasus ini berhasil.

3) Kasus Gambar dan Pesan Hasil Tes COVID-19 Tidak Valid

Kasus pengujian ini melibatkan *input* gambar dan pesan yang tidak valid. Seharusnya, hasil tes positif, namun pada gambar dituliskan hasil tes negatif sehingga *input* pesan juga negatif. Berdasarkan hasil pengujian yang telah dilakukan di atas, *output response payload* yang dihasilkan sesuai dengan ekspektasi. Artinya, pengujian untuk kasus ini berhasil.

V. KESIMPULAN DAN SARAN

Tanda tangan digital yang dibangun dari fungsi *hash* Keccak dan algoritma RSA dapat disisipkan pada *file* gambar menggunakan metode LSB untuk menjaga keaslian dan integritas data. Implementasi pembangkitan kunci, penandatanganan, serta verifikasi tanda tangan dapat memanfaatkan *Representational State Transfer Application Programming Interface* (REST API). Implementasi solusi ini terbukti berhasil dalam melakukan validasi keaslian dan integritas hasil tes COVID-19.

Untuk pengembangan di masa depan, hal ini dapat diimplementasikan dengan membuat standar yang seragam untuk seluruh instansi penyedia layanan tes COVID-19. Dengan demikian, hal ini dapat diimplementasikan secara luas dan mencegah penyebaran COVID-19. Harapannya, dengan pengembangan implementasi ini, pandemi COVID-19 dapat segera berakhir.

UCAPAN TERIMA KASIH

Terima kasih kepada Tuhan Yang Maha Esa, atas berkat dan rahmat-Nya makalah ini berhasil diselesaikan. Terima kasih juga kepada Dr. Ir. Rinaldi Munir, M.T. yang telah berbagi ilmu dan pengalamannya melalui mata kuliah II44031 Kriptografi dan Koding selama semester genap tahun ajaran 2021/2022. Terima kasih pula untuk asisten dosen yang suportif dan membantu dalam kegiatan kuliah. Kepada teman-teman peserta kuliah Kriptografi dan Koding yang telah bersama-sama melalui satu semester ini, terima kasih banyak. Semoga kita semua senantiasa diberi kesehatan dan kesuksesan.

REFERENSI

[1] Adhi, Dimas Bagas Satrio, "Pembuatan Sistem Pengamanan Hasil Tes Covid-19 Menggunakan Metode Digital Signature (Digest)", Institut Teknologi Sepuluh Nopember, Februari 2022

[2] Aptanagi, Pandyaka, "Implementasi Kriptografi Kunci Publik RSA dan Fungsi Hash SHA3 sebagai *Digital Signature* pada Pembayaran Digital", Institut Teknologi Bandung, Desember 2020

[3] Riyadi, Inka Anindya, "Implementasi Digital Signature pada Bukti Transfer menggunakan Kriptografi Kunci Publik RSA, Fungsi Hash SHA-256, dan Steganografi", Institut Teknologi Bandung, Desember 2021

[4] Munir, Rinaldi. 2022. Slide Kuliah II4031 Kriptografi dan Koding: Steganografi

[5] Munir, Rinaldi. 2022. Slide Kuliah II4031 Kriptografi dan Koding: Kriptografi Kunci-Publik (Public-key Cryptography)

[6] Munir, Rinaldi. 2022. Slide Kuliah II4031 Kriptografi dan Koding: Algoritma RSA

[7] Munir, Rinaldi. 2022. Slide Kuliah II4031 Kriptografi dan Koding: Fungsi hash

[8] Munir, Rinaldi. 2022. Slide Kuliah II4031 Kriptografi dan Koding: Fungsi hash SHA-3 (Keccak)

[9] Munir, Rinaldi. 2022. Slide Kuliah II4031 Kriptografi dan Koding: Tanda-tangan digital (digital signature)

PERNYATAAN

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

Bandung, 25 Mei 2022

Nadya Laurentia
18219071