Implementasi Algoritma RSA dan SHA3 dalam Pembuatan Web Token

Eka Novendra Wahyunadi / 13517011
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13517011@std.stei.itb.ac.id

***Abstrak*—Perkembangan teknologi web semakin pesat, salah satunya *web service* yang penggunaannya terus meningkat. Oleh karena itu perlu adanya peningkatan keamanan pada *web service*. Cara yang umum digunakan untuk meningkatkan keamanan pada *web service* adalah dengan penambahan fitur autentikasi dan otorisasi dengan memanfaatkan *token*. Pada makalah ini akan dibahas mengenai autentikasi dan otorisasi pada *web service* dengan *token* yang dibuat dengan menggunakan algoritma RSA dan SHA3.**

***Kata Kunci*—RSA, SHA3, Web Token, Autentikasi, Otorisasi**

I. PENDAHULUAN

Saat ini perkembangan teknologi web cukup pesat karena semakin banyak penggunanya. Oleh karena itu, perkembangan ini perlu diiringi dengan perkembangan pada teknologi pengamanan web. Salah satu teknologi web yang paling sering digunakan adalah *web service*, sehingga teknologi pengamanan sebuah *web service* menjadi sangat penting. Apabila sebuah *web service* tidak dapat melindungi data penggunanya, maka orang akan enggan untuk menggunakan *web service* tersebut.

Salah satu upaya untuk melindungi sebuah *web service* adalah dengan menambahkan fitur autentikasi dan otorisasi pada *web service* tersebut. Autentikasi adalah proses untuk membuktikan kebenaran identitas dari pengguna, sedangkan otorisasi adalah proses pemberian akses kepada pengguna yang telah melakukan autentikasi sesuai dengan hak dan kewenangan yang dimiliki oleh pengguna tersebut. Autentikasi dapat dilakukan dengan menggunakan data yang dapat dibuktikan keasliannya, seperti *email*, *password*, pin, dan sebagainya. Autentikasi dan otorisasi dibutuhkan agar tidak sembarangan orang bisa menggunakan *web service* tersebut, selain itu autentikasi dan otorisasi juga dapat digunakan untuk mencegah pengaksesan data pengguna oleh pihak tidak berwenang dan kebocoran data.

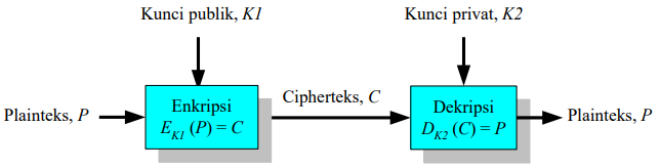
Fitur autentikasi dan otorisasi yang paling umum digunakan adalah dengan pemberian *token* pada pengguna. *Token* adalah sebuah *string* yang dapat dipergunakan sebagai penanda bahwa seorang pengguna terdaftar pada suatu *web service* dan berhak untuk mempergunakan *web service* tersebut. Pada sebuah token umumnya terdapat identitas dari pengguna beserta sebuah *digital signature* yang dapat dipergunakan untuk membuktikan keaslian dari *token* tersebut. Dengan adanya digital signature ini pengguna tidak dapat secara sembarangan membuat token sendiri atau mengubah informasi yang ada pada suatu token agar dapat dipergunakan untuk mengakses sebuah web service.

Pada makalah ini, akan dilakukan pembahasan mengenai autentikasi dan otorisasi sebuah *web service* dengan menggunakan *token* yang dibuat dengan memanfaatkan algoritma RSA dan SHA3 untuk membuat *digital signature* pada *token* tersebut.

II. DASAR TEORI

A. Kriptografi Kunci-Publik

Salah satu masalah yang ditemui pada sistem kriptografi kunci-simetri adalah bagaimana mengirimkan kunci rahasia kepada penerima pesan. Oleh karena itu muncullah ide kriptografi kunci-publik pada tahun 1976 [1]. Ide dari sistem kriptografi kunci-publik ini adalah pengirim dan penerima yang memiliki sepasang kunci, yaitu kunci publik dan kunci privat. Kunci publik digunakan untuk mengenkripsi pesan dan kunci privat digunakan untuk mendeskripsi pesan. Dengan sistem seperti ini kebutuhan untuk mengirim kunci rahasia tidak lagi ada. Contoh algoritma kriptografi kunci-publik adalah algoritma ElGamal dan algoritma RSA. Ilustrasi yang menunjukkan cara kerja sistem ini secara umum dapat dilihat pada gambar 1.



Gambar 1. Ilustrasi kriptografi kunci-publik [1]

B. RSA

RSA adalah salah satu algoritma kriptografi kunci-publik. Algoritma RSA adalah sebagai berikut [2].

Pembangkitan pasangan kunci

1. Pilih dua bilangan prima sembarang, misal a dan b. Jaga kerahasiaan a dan b ini.
2. Hitung n = ab, besaran n tidak perlu dirahasiakan.

3. Hitung m = (a-1)(b-1). Sekali m telah dihitung, a dan b dapat dihapus untuk mencegah diketahuinya oleh pihak lain.
4. Pilih sebuah bilangan bulat untuk kunci publik, misal e, yang relatif prima terhadap m.
5. Hitung kunci dekripsi, misal d, dengan kekongruenan ed ≡ 1(mod m).

Proses Enkripsi

1. Nyatakan pesan menjadi blok – blok plainteks: p_1 , p_2 , p_3 , … (harus dipenuhi persyaratan bahwa nilai p_1 harus terletak dalam himpunan nilai 0, 1, 2, 3, …, n-1 untuk menjamin hasil perhitungan tidak berada di luar himpunan).
2. Hitung blok cipherteks c_1 untuk blok plainteks p_1 dengan persamaan
$$c_i = p_i^e \ mod \ n$$
yang dalam hal ini, e adalah kunci publik.

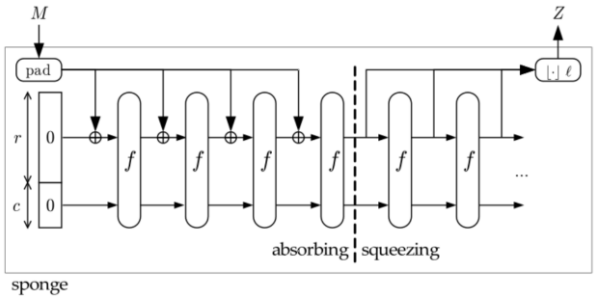
Proses Dekripsi

1. Proses dekripsi dilakukan dengan menggunakan persamaan
$$p_i = c_i^d \ mod \ \text{n}$$
yang dalam hal ini, d adalah kunci pribadi.

Dalam implementasinya, nilai a dan b disarankan nilai yang sangat besar (200 angka) agar pekerjaan memfaktorkan n menjadi faktor primanya menjadi sangat sulit.

C. SHA3

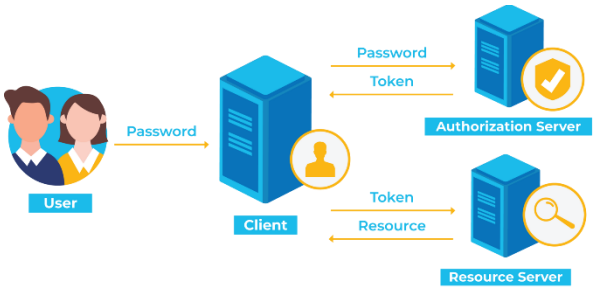
SHA3 atau Keccak merupakan sebuah fungsi hash yang menggunakan konstruksi 'spons' (*sponge construction*). Keccak menggunakan fungsi non kompresi untuk menyerap dan kemudian memeras *digest*. SHA3 memiliki dua fase, yaitu fase penyerapan dan fase pemerasan [3]. Ilustrasi konstruksi spons yang dimiliki oleh SHA3 dapat dilihat pada gambar 2.



Gambar 2. Ilustrasi konstruksi spons SHA3 [3]

D. Autentikasi Berbasis *Token*

Autentikasi berbasis *token* adalah sebuah protokol yang memungkinkan pengguna untuk memverifikasi identitas mereka, dan sebagai gantinya menerima *token* yang dipergunakan untuk akses. Selama masa pakai *token*, pengguna dapat mengakses *web service* tempat *token* diterbitkan, daripada harus memasukkan kembali kredensial setiap mereka menggunakan web service yang dilindungi *token* yang sama [4]. Ilustrasi protokol ini dapat dilihat pada gambar 3.



Gambar 3. Ilustrasi autentikasi berbasis token [4]

III. IMPLEMENTASI

A. Struktur Token

Struktur dari token yang akan dibuat terdiri dari dua bagian, yaitu bagian "payload" dan bagian "signature". Kedua bagian ini akan disambungkan dengan bantuan tanda titik. Sehingga format hasil token yang dibuat adalah sebagai berikut.

```
ewogIGVtYWlsOiDigJhleGFtcGxlQGV4YW
1wbGUuY29t4oCZLAogIHBhc3N3b3JkOiDi
gJhwYXNzd29yZOKAmQp9Cg==.Ddbc9sGrv
JrIWQSI%2FiFrU0vXCkYeAOdhE%2BmvajH
JlG3MJ0mvD9frPCOhEd%2BunokSY4R2Lhw
CdeK%2BsW%2BHCFTjRdANE3PnjnTAtTt94
c9FSpsEegQyxZGqXxttooVlSS6kSmXb3Xm
xpkPHmkK0mccpJigjfy6VQqMijwnv5xfIk
ExdVTkwjK%2BFMFFN8n%2B466iys7f9Bct
turxuVfegJMxGNZO%2ByJBUYZDDxlmCrVC
tQTUPJG0SVvYcUdtyVD2CFwzh
```

Payload akan berisi data yang akan dimasukkan ke dalam token, contohnya seperti identitas pengguna, waktu token dibuat, dan waktu token kedaluwarsa. Payload berbentuk JSON (JavaScript Object Notation) dan dimasukkan ke dalam token dalam bentuk base64. Payload digunakan juga pada pembuatan signature. Berikut merupakan contoh payload.

```
{
  email: 'example@example.com',
  password: 'password'
}
```

Signature berisi data payload yang dienkripsi dengan menggunakan algoritma RSA setelah dilakukan hash dengan menggunakan SHA3. Tujuan dari hash pada payload sebelum dienkripsi adalah agar data tidak dapat diambil kembali dari signature, dan agar ukuran dari signature bisa menjadi lebih kecil.

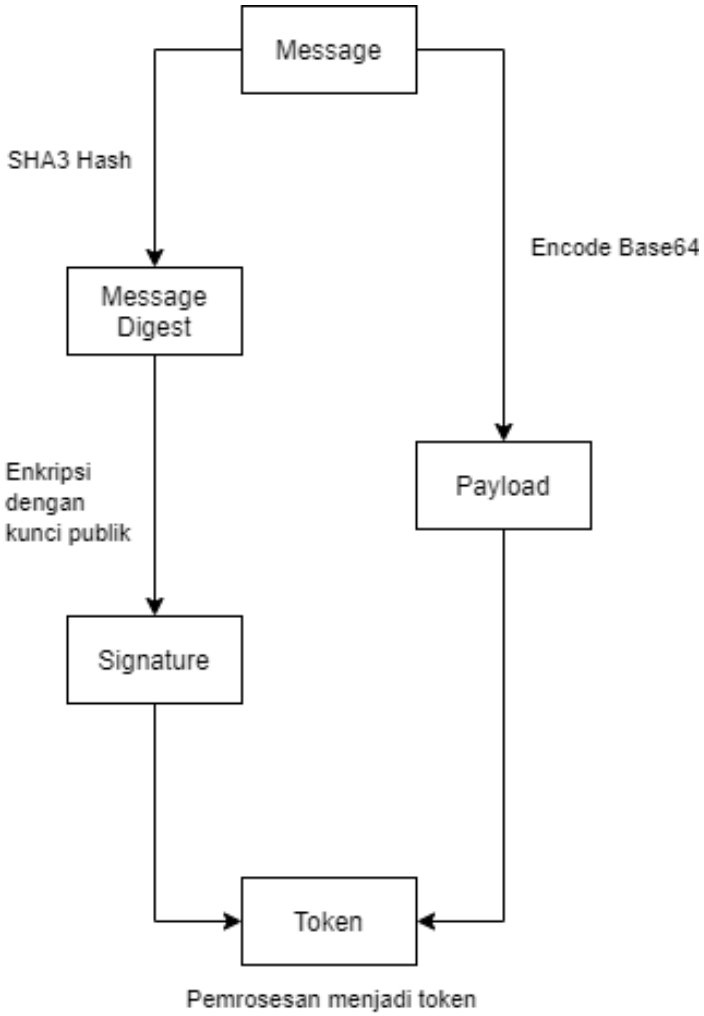
B. Pembuatan *Token*

Pembuatan token diawali dengan mendapatkan kunci publik dan kunci privat terlebih dahulu. Kunci yang digunakan dapat berasal dari kunci yang sudah dibuat sebelumnya atau dengan membuat pasangan kunci baru. Kunci privat perlu disimpan oleh server, sedangkan untuk kunci publik dapat dikembalikan bersama token jika suatu

saat ingin membuat token kembali dengan pasangan kunci yang sama.

Setelah mendapatkan pasangan kunci, message akan diproses menjadi *token*. Pertama message di *encode* menjadi base64 sehingga didapatkan bagian payload untuk token. Selanjutnya message di hash dengan SHA3 sehingga didapatkan message digest. Message digest yang sebelumnya didapatkan dienkripsi dengan menggunakan kunci publik, sehingga didapatkan bagian signature dari *token*.

Setelah bagian payload dan signature didapatkan, kedua bagian tersebut disambungkan dengan bantuan tanda titik dan dikembalikan sebagai *token*. Alur pembuatan *token* dapat dilihat pada gambar 3.

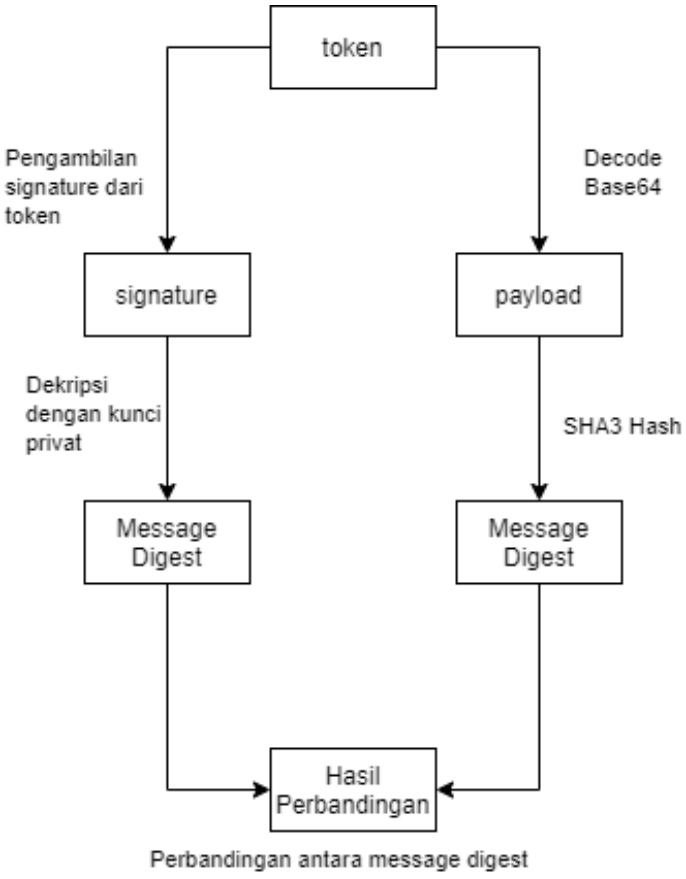


Gambar 3. Alur pembuatan token

C. Verifikasi *Token*

Verifikasi *token* diawali dengan mendapatkan terlebih dahulu kunci privat yang sebelumnya sudah disimpan. Selanjutnya token dipisahkan menjadi dua bagian yaitu payload dan signature dengan bantuan tanda titik yang sebelumnya digunakan untuk menyambung kedua bagian tersebut. Untuk bagian payload dilakukan *decode* dari bentuk base64 ke bentuk JSON, lalu dilakukan hash dengan SHA3 pada payload tersebut sehingga didapatkan message digest dari payload. Setelah itu signature didekripsi dengan menggunakan kunci privat, sehingga didapatkan message digest dari signature.

Setelah kedua message digest didapatkan, dilakukan perbandingan dari message digest tersebut. Jika kedua message digest tersebut sama, maka *token* dapat dipastikan valid dan pengguna berhak untuk mengakses *web service* tersebut. Alur verifikasi token dapat dilihat pada gambar 4.



Gambar 4. Alur verifikasi token

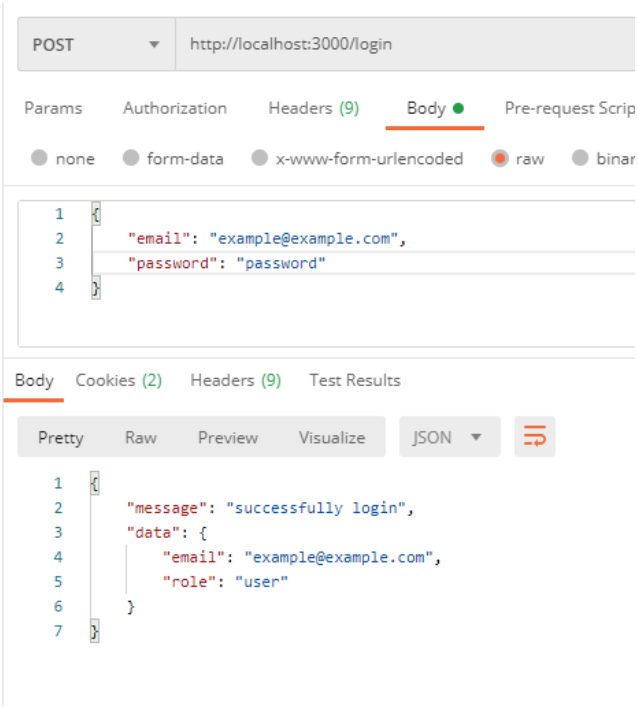
IV. HASIL PENGUJIAN DAN ANALISIS

A. Pengujian Pembuatan *Token*

Untuk mengetahui apakah *token* dapat digunakan pada sebuah *web service*, dilakukan pengujian dengan data sebagai berikut.

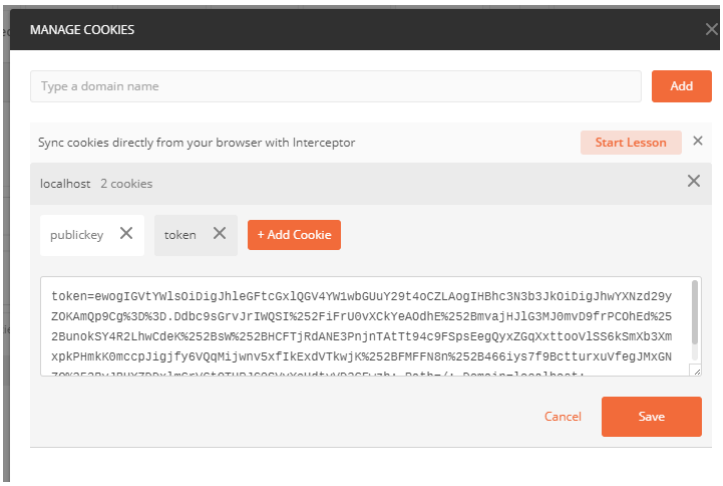
```
{
  email: 'example@example.com',
  password: 'password'
}
```

Didapatkan response seperti pada gambar 5.



Gambar 5. Response saat pembuatan *token*

Dari gambar tersebut terlihat bahwa dikembalikan juga dua buah cookie, detail cookie tersebut dapat dilihat pada gambar 6.

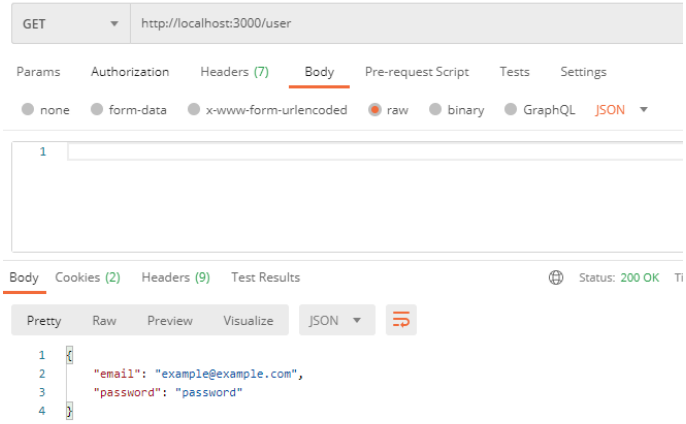


Gambar 6. Detail cookie yang diterima

Dari detail cookie pada gambar 6 terlihat publickey dari pembuatan *token* dan *token* berhasil dibuat dan disimpan pada client.

B. Pengujian Verifikasi *Token*
Dilakukan pengujian untuk mengetahui apakah token berhasil diverifikasi atau tidak, dan apakah token bisa terdeteksi jika tidak valid.

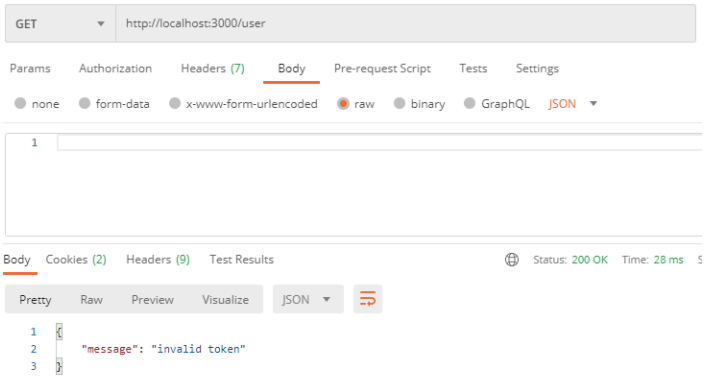
1. Pengujian *token* valid
Hasil pengujian untuk *token* valid dapat dilihat pada gambar 7.



Gambar 7. Hasil pengujian untuk *token* valid

Dari gambar 7 terlihat bahwa verifikasi *token* berhasil dan data pengguna berhasil diambil.

2. Pengujian *token* tidak valid
Hasil pengujian untuk *token* tidak valid dapat dilihat pada gambar 8.



Gambar 8. Hasil pengujian untuk token *tidak* valid

Dari gambar 8 terlihat bahwa verifikasi *token* tidak berhasil dan server mengembalikan pesan bahwa *token* tidak valid.

V. KESIMPULAN

Algoritma RSA dan fungsi hash SHA3 dapat diimplementasikan dalam bentuk pembuatan *web token* sebagai mekanisme autentikasi dan otorisasi pada *web service*. Pembuatan dan verifikasi *token* berhasil dilakukan. Selain itu *token* yang tidak valid juga berhasil terdeteksi. Kedepannya pengembangan untuk peningkatan mekanisme autentikasi dan otorisasi ini masih dapat dilakukan.

VI. UCAPAN TERIMA KASIH

Puji syukur ke hadirat Tuhan yang Maha Esa, karena atas berkat dan rahmat-Nya penulis memiliki kesempatan untuk menyelesaikan makalah ini. Penulis juga mengucapkan terima kasih kepada orang tua penulis atas dukungannya dalam penulisan makalah ini. Tak lupa penulis juga mengucapkan terima kasih kepada semua pihak yang telah membantu dalam penulisan makalah ini, terutama Bapak Rinaldi Munir yang telah memberikan wawasan dan pengetahuan mengenai kriptografi, serta teman-teman penulis yang senantiasa memberikan saran dan dukungan sehingga penulis dapat menyelesaikan makalah ini.

REFERENSI

[1] R. Munir, “Kriptografi Kunci-Publik,” 2020, [Online]. Available: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kriptografi-Kunci-Publik-2020.pdf.

[2] R. Munir, *Matematika Diskrit*, 4th ed. Penerbit INFORMATIKA Bandung, 2010.

[3] R. Munir, “SHA-3 (Keccak),” vol. 3, [Online]. Available: http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/SHA-3-2020.

[4] Okta Inc., “What Is Token-Based Authentication?” https://www.okta.com/identity-101/what-is-token-based-authentication/ (accessed Dec. 21, 2020).

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, 21 Desember 2020

Eka Novendra Wahyunadi
13517011