Implementasi Algoritma RSA dan Hash SHA-3 untuk Aplikasi Voting

Stephanie Hutagalung 18220001
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): stephaniehutagalung8@gmail.com

***Abstract*— Dalam organisasi, umumnya, untuk memperoleh suara terbanyak dari anggota, salah satu caranya adalah dengan melakukan voting. Dalam melakukan voting ada beberapa hal yang perlu diperhatikan. Salah satunya adalah kerahasiaan dan persentase keputusan dari voting. Untuk mempermudah pelaksanaan voting diperlukan aplikasi voting yang menjaga kerahasiaan dan memastikan pemilih memilih sesuai dengan ketentuan yang ada. Karena itu, diperlukan aplikasi voting yang mengimplementasikan kunci publik RSA dan Hash SHA-3 untuk membangkitkan token dari data pemilih untuk menjamin kerahasiaan data pemilih. Implementasi algoritma yang digunakan sudah berhasil membangkitkan token dan mengolah data vote dari pemilih.**

***Keywords*— Token; RSA; SHA-3; Aplikasi Voting;**

I. PENDAHULUAN

Organisasi merupakan suatu sistem yang terdiri dari bagian - bagian yang saling berkaitan satu dengan yang lainnya. Organisasi haruslah memiliki anggota, tujuan, dan pemimpin. Beberapa organisasi membuat keputusan secara demokrasi, yaitu dengan pengambilan keputusan dengan cara suara terbanyak. Untuk memperoleh suara terbanyak dari anggota, salah satu caranya adalah dengan melakukan voting. Umumnya, pengambilan keputusan dilakukan ketika memilih pemimpin baru, memutuskan peraturan baru, dan lainnya.

Kegiatan voting sering dilakukan karena akan mempersingkat waktu untuk mengambil keputusan atau menyelesaikan masalah. Selain itu juga suara yang terbanyak menghasilkan keputusan yang mutlak dan tidak dapat dibantah oleh pihak manapun. Voting juga dapat menghemat pikiran atau tenaga dari tenaga pekerja diskusi karena akan mempermudah untuk mengeliminasi hal-hal yang tidak sesuai dengan keingingan banyak orang.

Namun, kegiatan voting juga memiliki kekurangan. Aspirasi dari semua pesereta tidak sepenuhnya disalurkan karena aspirasi yang minoritas akan dieliminasi. Keputusan dari voting juga tidak hasil dari mufakat. Akan ada beberapa pihak yang tidak dapat menerima keputusan akhir atau beberapa pihak terpasa untuk menerima keputusan akhir tersebut.

Dalam melakukan voting ada beberapa hal yang perlu diperhatikan. Salah satunya adalah kerahasiaan. Anggota dapat menyampaikan suaranya tanpa diketahui oleh pihak manapun. Pemilih dapat memberikan suara dengan tidak diketahui oleh orang lain apapun pilihannya. Selain itu, dalam voting juga perlu memperhatikan jumlah anggota yang melakukan voting. Keputusan dari voting dapat dianggap sah jika setengah dari anggota menyetujui salah satu hasil dari voting tersebut.

Beberapa cara voting yang biasa dilakukan adalah dengan mengumpulkan anggota di satu tempat pada waktu tertentu. Kemudian, mengambil keputusan dengan mengangkat tangan. Keputusan akhir diambil berdasarkan jumlah orang yang mengangkat tangan. Selain itu, dapat juga dilakukan dengan menggunakan kotak suara. Setiap anggota memilih dengan menuliskannya pada kertas lalu mengumpulkannya di suatu kotak tertentu.

Saat ini, organisasi tidak hanya dibatasi oleh tempat. Anggota dari suatu organisasi bisa saja berada di tempat yang berbeda satu sama lain atau memiliki kesibukan yang berbeda dalam satu waktu. Hal ini merupakan tantangan untuk melakukan voting. Karena itu perlu dilakukan voting menggunakan suatu aplikasi untuk mempermudah pengumpulan suara yang dapat dilakukan di dimana pun dan kapan pun serta menghemat sumber daya.

Aplikasi voting yang akan digunakan adalah teknologi yang mengimplementasikan kunci publik RSA dan Hash SHA-3 untuk membangkitkan token dari data pemilih. Dengan dikembangkannya aplikasi voting ini diharapkan proses pemungutan suara dapat dijamin kerahasiaannya.

II. DASAR TEORI

A. Algoritma Kunci Publik RSA

Algoritma RSA (Rivest-Shamir Adleman) adalah algoritma yang dibuat oleh Ronald Rivest, Adi Shamir, dan Leonard Adleman, pada tahun 1976 yang merupakan peneliti dari MIT. Algoritma RSA ini merupakan algoritma kunci-publik yang saat ini paling terkenal dan paling banyak diaplikasikan.[2]

Berikut ini merupakan property dari algoritma RSA.

1. p dan q bilangan prima (rahasia)

2. $n = p \cdot q$ (tidak rahasia)
3. $\phi(n) = (p - 1)(q - 1)$ (rahasia)
4. e (kunci enkripsi) (tidak rahasia)

   e merupakan bilangan bulat yang dipilih sebagai kunci public. Syarat: PBB(e, $\phi(n)$) = 1 , PBB = pembagi bersama terbessar = gcd
5. d (kunci dekripsi) (rahasia)

   d dihitung dari $d \equiv e^{-1} \bmod (\phi(n))$ dan memenuhi persamaan $ed \equiv \bmod (\phi(n))$
6. m (plainteks) (rahasia)

   informasi yang dirahasiakan.
7. c (cipherteks) (tidak rahasia)

Dari komponen RSA di atas, pasangan kunci privat adalah (d, n) dan pasangan kunci publik adalah (e, n). Proses enkripsi dapat dilakukan dengan menbaginya menjadi blok-blok lebih kecil jika memungkinkan. Kemudian, cipherteks, c, dihitung untuk plainteks, m, menggunakan kunci public, e, dengan persamaan berikut.

$$c = m^e \bmod n$$

Sementara itu proses dekripsinya adalah dengan menghitung blok plainteks, m, dengan blok chiperteks, c, dengan menggunakan kunci privat , d, dengan persamaan berikut.

$$m = c^d \bmod n$$

B. *Algoritma Hash SHA-3*

Algoritma hash adalah fungsi yang menerima pesan dengan Panjang yang sembarang kemudian melakukan kompresi menjadi digest dengan panjang yang tetap. Beberapa variasi Panjang digest adalah 1268 bit, 224 bit, 256 bit, dan 512 bit.

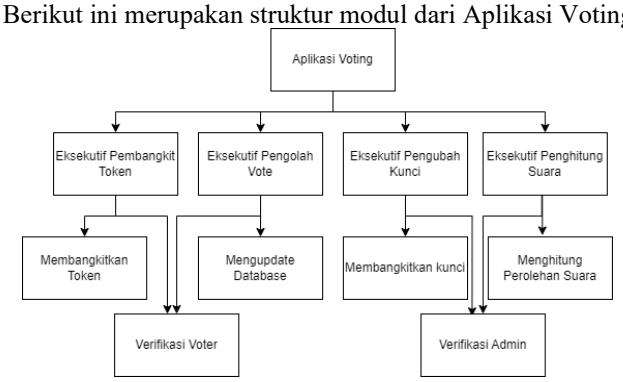
Hash SHA-3 atau yang juga dikenal sebagai keccak. Algoritma ini merupakan didesain oleh Guido Breton, Joan Daemen, Michaël Peeters dan Gilles Van Assche. Algoritma ini merupakan algoritma yang terpilih sebagai SHA-3 pada kompetisi terbuka yang diselenggarakan *National Institute of Standards and Technology*. Algoritma ini dipilih karena berbeda dari finalis lainnya dalam hal menggunakan konstruksi 'spons'. Algoritma ini menggunakan fungsi non-kompresi untuk menyerap dan kemudaian memeras digest. Karena berbeda dengan pendekatan lainnya inilah, Keccak dipilih sebagai algoritma SHA-3 yang merupakan komplementasi dari SHA-1 dan SHA-2.[1]

III. DESAIN DAN RANCANGAN

Sistem yang akan dikembangkan akan dapat membangkitkan token dari data pemilih dengan menggunakan algoritma kunci publik RSA dan algoritma Hash SHA-3.

A. Perancangan Sistem Aplikasi Voting

Berikut ini merupakan struktur modul dari Aplikasi Voting



Gambar 3.1 Struktur Modul Aplikasi Voting

Aplikasi voting akan terdiri dari eksekutif pembangkit token, eksekutif pengolah vote, eksekutif pengubah kunci, dan Eksekutif penghitung suara. Eksekutif pembangkit kunci akan memverifikasi *voter* dengan meminta masukkan NIM dan password. Jika NIM dan password tersebut sesuai dengan yang ada di database, maka sistem akan memberikan token yang merupakan enkripsi dari hash data *voter.*

Eksekutif pengolah vote juga akan memverifikasi *voter* dengan menerima masukan NIM, password, dan token. *Voter* juga akan diminta pilihannya yang akan diubah di database. Hanya masukan NIM, password, dan token yang sesuai yang akan berhasil melakukan vote. Pengolahan vote juga akan mengubah status dari voter. Status ini menunjukkan apa voter tersebut sudah melakukan pemilihan atau tidak. Hal ini dilakukan untuk memastikan setiap voter hanya memilih sebanyak satu kali.

Eksekutif pengubah kunci akan memverifikasi admin dengan menerima masukkan username dan password. Jika sesuai, maka sistem akan mengubah kuncinya secara otomatis.

Eksekutif penghitung suara juga akan memverifikasi admin dengan menerima masukan username dan password. Jika sesuai, maka sistem akan menampilkan perolehan suara terbanyak dan persentasenya.

Verifikasi *voter* dan verifikasi admin perlu dilakukan agar hanya anggota yang berhak melakukan pemilihan saja yang akan memilih dan hanya admin saja yang dapat mengubah kunci yang digunakan sistem dan menampilkan hasil perolehan suara.

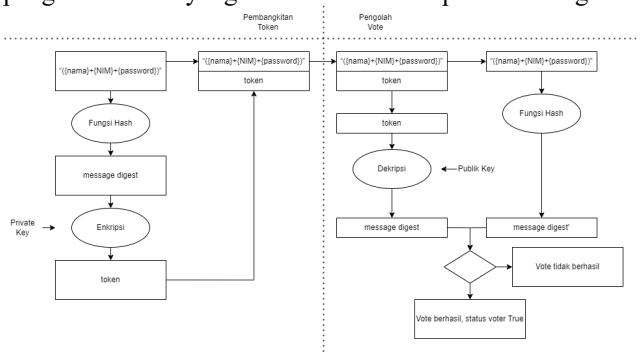
B. *Perancangan Modifikasi Algoritma Enkripsi Data Pemilih*

Algoritma enkripsi data pemilih menjadi token dilakukan pada pembangkitan token. Data yang akan dienkripsi adalah

nama_nim_pass = "({nama}+{NIM}+{password})"

Dengan {nama} adalah nama dari *voter*, {NIM} adalah NIM dari *voter*, dan {password} adalah password dari *voter*. Pembangkitan token ini hampir sama dengan cara kerja pembangkitan tanda tangan digital. Token yang sudah dienkripsi akan didekripsi untuk mengubah status dari *voter* dan mengolah data vote di database.

Berikut merupakan proses pembangkitan token dan pengolahan vote yang dilakukan sistem aplikasi voting.



Gambar 3.2 Pembangkitan Token dan Pengolahan Vote

Pada pembangkitan token, nama_nim_pass akan di-hashing dengan menggunakan algoritma hash SHA-3 yang akan menghasilkan message digest. Kemudian message digest akan dienkripsi dengan private key yang telah disimpan pada sistem sebelumnya. Hasil dari enkripsi message digest adalah token.

Pada pengolahan vote, sstem akan melakukan hashing pada nama_nim_pass yang akan menghasilkan message digest. Sementara itu, token juga akan didekripsi dengan menggunakan public key yang akan menghasilkan message digest. Jika kedua message digest sama, sistem akan menyimpan hasil vote dan mengubah status voter menjadi True yang berarti voter telah melakukan pemilihan, voting berhasil. Namun, jika message digest tidak sama, voting tidak berhasil.

C. *Perancangan Pembangkitan Kunci dengan Algoritma RSA.*

Pembangkitan kunci dengan algoritma RSA tidak mengalami modifikasi. Sebelumnya sistem akan diberikan konstanta sebagai private key, public key, dan nilai n dari sistem. Namun, admin dapat mengubah key untuk menjaga kerahasiaan dari token voter.

D. *Batasan Perancangan.*

Dalam perancangan aplikasi voting ini terdapat beberapa Batasan.

1. Aplikasi voting yang dirancang merupakan aplikasi desktop.
2. Data voter diakses secara langsung ke database.
3. Data candidate (pilihan yang devoting) diakses secara langsung ke database.
4. Candidate yang disediakan hanyalah 2 yaitu, candidate1 dan candidate2
5. Voter hanya dapat melakukan voting sebanyak satu kali.
6. Mengubah kunci dan menghitung hasil peroleh suara hanya dapat dilakukan oleh admin.

IV. IMPLEMENTASI PROTOTIPE

Berikut ini adalah proses implementasi dari desain dan rancangan yang dibuat pada bagian sebelumnya. Implementasi akan dilakukan pada aplikasi voting adalah dengan menggunakan bahasa pemrograman Python. Bahasa pemrograman Python dipilih pada implementasi ini karena kemudahan dalam pengembangan serta tersedianya banyak sumber daya online sebagai bahan referensi. Dalam membuat antarmuka sistem akan digunakan library Tkinter. Library Tkinter ini dipilih karena kemudahan pengembangan antarmuka.

A. *Implementasi Pembangkitan Kunci dengan Algoritma RSA*

Berikut ini merupakan implementasi dari fungsi pembangkitan kunci public key dan private key dengan algoritma RSA.

```
# Menu pembangkitan kunci publik dan kunci
privat RSA
import random

def isPrime(number):
    if(number < 2):
        return False
    else:
        for i in range(2, number):
            if((number % i)) == 0:
                return False
        return True

def primeNumber(minRange, maxRange):
    while True:
        number = random.randint(minRange,
maxRange)
        if isPrime(number):
            return number

def gcd(a, b):
    while b != 0:
        a, b = b, a % b
    return a

def eValue(totient):
    while True:
        e = random.randint(1, totient)
        if(gcd(e, totient)==1):
            break
    return e

def dValue(e, totient):
    d = 2
    while(1):
```

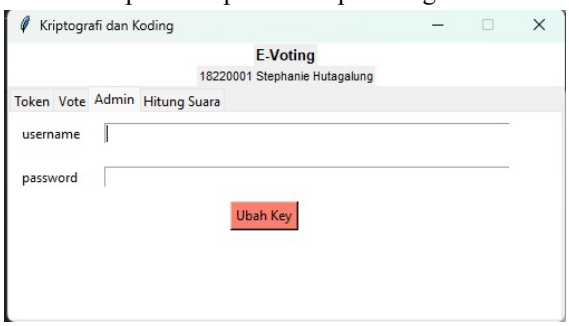
```
        if((d*e) % totient) == 1:
            break
        d += 1
    return d

def genPubPrivKey():
    p = primeNumber(100, 500)
    q = primeNumber(100, 500)

    if (p != q):
        n = p*q
        totient = (p-1)*(q-1)
        # PUBLIK KEY
        e = eValue(totient)
        # PRIVATE KEY
        d = dValue(e, totient)
        return (e, d, n)
```

```
EnkripDekrip.enkrip(nama_nim_pass, self.d,
self.n)
                self.StringToken.set(f"Token
adalah {Token}")

messagebox.showinfo("Token",f"Token adalah
{Token}")
            else:

messagebox.showerror("Error","NIM dan
password salah")
        else:
            messagebox.showerror("Error","NIM
dan password salah")
    except:
        messagebox.showerror("Error","NIM dan
password salah")
```

Berikut ini merupakan public key dan private key yang telah ditetapkan sebagai konstantas dari sistem.

```
self.d = 26831
self.n = 105559
self.e = 40271
```

Berikut ini merupkan tampilan GUI pembangkitan kunci.



Gambar 4.1 Tampilan GUI pembangkitan kunci

B. Implementasi Pembangkitan Token

Berikut ini merupakan implementasi pembangkitan token.

```
#  Generate Token
def GenerateToken(self):
    NIM=''
    password = ''
    try:
        NIM = int(self.nim_voter.get("1.0",
"end-1c"))
        password = self.password.get("1.0",
"end-1c")
        if (NIM,) in self.getnimvoter():
            if self.checkpassword(NIM,
password):
                nama = self.getnamavoter(NIM)
                nama_nim_pass =
str(f"({nama}+{NIM}+{password})")
                Token =
```

Berikut ini merupakan implementasi dari hash dengan menggunakan algoritma SHA3 dan enkripsi.

```
from hashlib import sha3_256

# Menghitung nilai hash dari pesan M
def Hash(message):
    hash = sha3_256(message.encode("latin-
1")).hexdigest()
    hashed = int(hash, 16)
    return hashed

# Mengenkripsi h dengan kunci privatnya (SK)
menggunakan persamaan enkripsi RSA, hasilnya
adalah signature S:
def enkrip(message,PK, n):
    hashed = Hash(message)
    S = (hashed**PK) % n
    return S
```

Berikut ini merupkan tampilan GUI pembangkitan token.

Gambar 4.2 Tampilan GUI pembangkitan token

C. Implementasi Pengembangan Pengolahan Vote

Berikut ini merupakan implementasi dari pengembangan pengolah vote.

```
def Vote(self) :
    try:
        NIM = int(self.nim_voter2.get("1.0",
"end-1c"))
        password = self.password2.get("1.0",
"end-1c")
        token = int(self.token.get("1.0",
"end-1c"))
        nama = self.getnamavoter(NIM)
        nama_nim_pass =
str(f"({nama}+{NIM}+{password})")
        if self.checkstatus(NIM):
            messagebox.showinfo("info", "Akun
sudah melakukan vote, tidak ada percobaan
voting kedua.")
        else:
            verifying =
EnkripDekrip.dekrip(nama_nim_pass, self.e,
self.n, token)
            if verifying:
                hasilvote =
int(self.radioe3.get())
                self.insertvote(hasilvote,
NIM)
                messagebox.showinfo("info",
"Voting berhasil!")
            else:

messagebox.showerror("Error","NIM, password,
dan token salah")
    except:
        messagebox.showerror("Error","NIM,
password, dan token salah")
```

Berikut ini merupakan implementasi fungsi dekripsi.

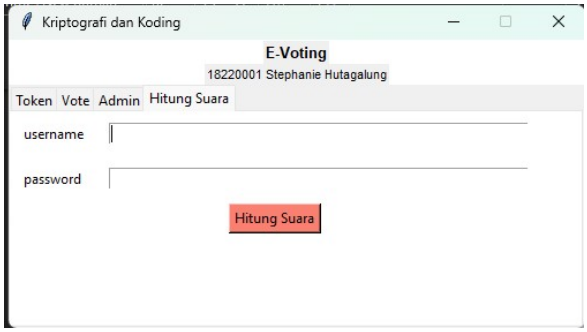
```
# dekripsi terhadap tanda-tangan S dengan
kuncipublik si pengirim (PK) menggunakan
persamaan dekripsi RSA:
def dekrip(message, PK, n, S):
    hashed = Hash(message)%n
    h= (S**PK)%n
    if int(h) == int(hashed):
        return True
    else:
        return False
```

Berikut ini adalah tampilan GUI dari pengolahan vote.

Gambar 4.3 Tampilan GUI pengolahan vote

D. Implementasi Penghitungan Suara

Berikut ini adalah tampilan GUI dari penghitungan suara.



Gambar 4.4 Tampilan GUI penghitungan suara

V. PENGUJIAN

Aplikasi yang sudah dikembangkan akan diuji untuk memastikan dapat digunakan sesuai fungsinya. Pengujian yang akan dilakukan adalah pengujian fungsional.

Berikut ini merupakan pengujian pembangkitan token.

TABLE I. TABEL PENGUJIAN PEMBANGKITAN TOKEN

No	***Kasus Uji***	***Masukan (Input)***	***Harapan Hasil (Output)***	***Hasil Pengujian***
1	Input Valid	NIM: 18220500 Pass : Pass1	Token dibangkitakan	Token adalah 94328
2	Input Tidak Valid	NIM: 18220500 Pass : Pass0	Token tidak dibangkitakan	Pesan Error : NIM dan Password salah

Berikut ini merupakan pengujian pengolahan vote.

TABLE II. TABEL PENGUJIAN PENGOLAHAN VOTE

No	*Kasus Uji*	*Masukan (Input)*	*Harapan Hasil (Output)*	*Hasil Pengujian*
1	Input Valid	NIM: 18220500 Pass : Pass1 Token : 94328	Voting berhasil	Pesan Info : Voting berhasil!
2	Input Tidak Valid	NIM: 18220500 Pass : Pass0 Token : 94328 Vote : 1	Voting tidak berhasil	Pesan Error : NIM, Password dan Token salah
3	Input Tidak Valid	NIM: 18220500 Pass : Pass1 Token : 94328 Vote : 0	Voting tidak berhasil	Pesan Error : Voting tidak berhasil! Voter belum memilih.

Berikut ini merupakan pengujian pembangkitaa key.

TABLE III. TABEL PENGUJIAN PEMBANGKITAN KEY

No	*Kasus Uji*	*Masukan (Input)*	*Harapan Hasil (Output)*	*Hasil Pengujian*
1	Input Valid	username : admin password : admin	Pembangkitan Key berhasil	Pesan Info : Key telah diubah!
2	Input Tidak Valid	username : Admin password : admin	Pembangkitan Key tidak berhasil	Pesan Error : Username dan password salah

Berikut ini merupakan pengujian penghitungan suara.

TABLE IV. TABEL PENGUJIAN PENGHITUNGAN SUARA

No	*Kasus Uji*	*Masukan (Input)*	*Harapan Hasil (Output)*	*Hasil Pengujian*
1	Input Valid	username : admin password : admin	Pembangkitan Key berhasil	Pesan Info : Hasil Perolehan Suara Seri!
2	Input Tidak Valid	username : Admin password : admin	Pembangkitan Key tidak berhasil	Pesan Error : Username dan password salah

VI. KESIMPULAN DAN SARAN

Aplikasi voting telah berhasil dibuat membangkitkan token dari data voter, pengolahan vote, pembangkitan kunci, dan penghitungan suara dengan menggunakan algoritma RSA dan algoritma SHA3.

Dalam pengembangan ini, terdapat beberapa kekurangan yaitu tidak adanya login dan sign up untuk voter sehingga untuk penambahan data voter dan pengubahan password voter harus melalui pengubahan database secara langsung. Selain itu, password yang disimpan di database juga masih berupa karakter password. Seharusnya untuk menjaga kerahasiaan password, password yang disimpan adalah message digest dari password yang sudah di- hashing. Aplikasi yang dikembangkan saat ini berupa aplikasi desktop, aplikasi voting perlu dikembangkan agar dapat digunakan di manapun dan kapan pun adalah aplikasi berbasi web.

Selain itu, untuk pengembangan lanjutan, aplikasi yang sudah dibuat dapat dikembangkan dengan mengembangkan login dan signup voter untuk mendaftar dan mengubah password. Selain itu juga penyimpanan password berupa message digest dari hashing password.

VIDEO LINK AT YOUTUBE

https://youtu.be/WJWpJCMFn8c

SOURCE CODE

https://github.com/srnstephanie/Aplikasi-Voting.git

ACKNOWLEDGMENT

Saya mengucapakan terima kasih kepada Tuhan Yang Maha Esa karena atas karunianya penulis dapat menyelesaikan makalah yang berjudul “Implementasi Algoritma RSA dan Hash SHA-3 untuk Aplikasi Voting”. Saya juga mengucapkan terima kasih kepada segala teman-teman mahasiswa II4031 Kriptografi dan Koding, Semester II Tahun 2022/2023 yang telah membantu saya selama kegiatan belajar mengajar pada mata kuliah II4031 Kriptografi dan Koding. Secara khusus, saya juga berterima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. Saya berharap hasil makalah ini dapat menjadi sumber inspirasi untuk pihak lainnya yang akan mengeksplorasi dan mengembangkan masalah maupun solusi serupa.

REFERENSI

[1] R. Munir, “Fungsi Hash SHA-3 (Keccak),” 2023.

[2] R. Munir, “Algoritma RSA,” 2023.

[3] R. Munir, “Tanda Tangan Digital (Digital Siganture),” 2023.

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, 22 Mei 2023

Stephanie Hutagalung, 18220001