Implementasi Algoritme Okamoto-Uchiyama untuk Penyembunyian Data pada Basis Data Pemilihan Umum

Izharulhaq - 13518092
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail (gmail): 13518092@std.stei.itb.ac.id

Abstraksi—Semua sistem basis data memiliki risiko untuk diretas karena sangatlah sulit untuk menciptakan suatu sistem basis data yang tidak dapat diretas oleh siapapun. Apabila suatu basis data diretas, akan ada pihak-pihak yang tidak diinginkan yang berusaha untuk memanfaatkan informasi yang ada pada basis data tersebut sesuai dengan kepentingannya. Salah satu solusi agar informasi yang ada pada basis data tidak mudah diakses oleh publik adalah dengan mengenkripsi setiap informasi yang ada pada basis data tersebut. Pada makalah ini, akan dibahas proses pengenkripsian data pada basis data suara pemilihan umum dengan menggunakan algortime Okamoto-Uchiyama dan fungsi *hash* SHA-3.

Kata Kunci—Kriptografi kunci publik, enkripsi homomorfik, algoritme Okamoto-Uchiyama, *e-voting*, SHA-3

I. Pendahuluan

Dalam negara-negara yang menerapkan sistem demokrasi, misalnya Indonesia, pemimpin dari negara-negara tersebut, misalnya presiden, dipilih oleh dengan menggunakan pemilihan umum. Berbagai macam teknologi telah dikembangkan untuk membantu penyelenggaraan pemilihan umum tersebut, salah satunya adalah teknologi *e-voting* atau pemungutan suara dengan menggunakan teknologi digital.

Untuk menerapkan teknologi *e-voting* dalam pemilihan umum, maka sistem *e-voting* yang dibuat haruslah memiliki keamanan yang sangat kuat sehingga sistem tidak bisa diretas. Akan tetapi, tidak ada suatu sistem yang tidak dapat diretas oleh siapapun.

Apabila basis data pada pemilihan umum berhasil diretas, maka akan ada pihak-pihak yang berusaha untuk memanipulasi pemilihan umum sesuai dengan kepentingannya. Salah satu solusi untuk mencegah hal tersebut terjadi adalah dengan menyembunyikan informasi yang ada pada basis data dengan cara mengenkripsi isi dari basis data tersebut.

Pada makalah ini, akan dibahas proses enkripsi basis data suara pada pemilihan umum dengan memanfaatkan algoritme kriptografi kunci-publik Okamoto-Uchiyama dan fungsi *hash* SHA-3.

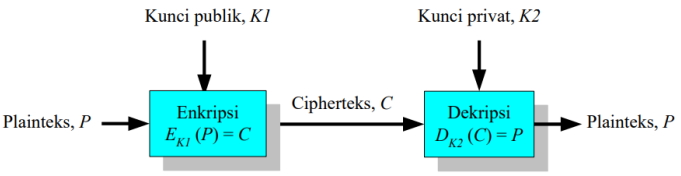
II. Landasan Teori

A. Kriptografi Kunci Publik

Hingga akhir tahun 1970, metode kriptografi yang paling umum digunakan adalah kriptografi simetri. Akan tetapi, kriptografi simetri memiliki kelemahan yang mengharuskan kedua belah pihak, baik pengirim maupun penerima pesan harus mengetahui kunci untuk enkripsi dan dekripsi.

Agar pengirim maupun penerima pesan dapat saling berbagi kunci, kunci harus dikirimkan melalui jalur yang aman. Akan tetapi, jalur tersebut umumnya lambat dan mahal. Untuk mengatasi masalah tersebut, muncullah ide kriptografi kunci publik pada tahun 1976 [1].

Pada kriptografi kunci publik, terdapat dua jenis kunci, yaitu kunci publik dan kunci privat. Pengirim pesan akan mengenkripsi pesan dengan menggunakan kunci publik milik penerima. Kemudian, setelah menerima pesan yang dienkripsi penerima akan mendekripsi pesan tersebut dengan menggunakan kunci privat miliknya. Berikut adalah ilustrasi yang menggambarkan proses enkripsi dan dekripsi pesan menggunakan kriptografi kunci publik.



Gambar 1 Proses dalam Kriptografi Kunci Publik [1]

B. Enkripsi Homomorfik

Apabila suatu plainteks harus diubah, maka cipherteks harus didekripsi terlebih dahulu menjadi plainteks untuk kemudian dilakukan modifikasi pada plainteks, dan terakhir dienkripsi kembali menjadi cipherteks yang baru. Pada cara tersebut, plainteks memiliki risiko tersadap oleh pihak yang tidak diinginkan [2]. Untuk menghindari risiko tersebut, muncullah enkripsi homomorfik.

Pada enkripsi homomorfik, komputasi perubahan yang diperlukan hanya dilakukan pada cipherteks tanpa harus mendekripsikan cipherteks terlebih dahulu. Pada saat cipherteks hasil modifikasi didekripsi, modifikasi yang sama juga akan terlihat pada plainteks [2].

Enkripsi homomorfik dapat bersifat aditif ataupun multiplikatif. Enkripsi homomorfik dikatakan bersifat aditif jika memenuhi persamaan

$$E(m_1 + m_2) = E(m_1) \otimes E(m_2) \quad (1)$$

dengan E adalah fungsi enkripsi, m_1 dan m_2 adalah plainteks yang akan dienkripsi, dan $\otimes$ adalah operasi (seperti +, ×, dan operasi lainnya) yang bergantung pada jenis enkripsi yang digunakan. Mirip halnya dengan sifat aditif, enkripsi homomorfik dikatakan bersifat multiplikatif jika memenuhi persamaan

$$E(m_1 \cdot m_2) = E(m_1) \otimes E(m_2) \quad (2)$$

dengan E adalah fungsi enkripsi, m_1 dan m_2 adalah plainteks yang akan dienkripsi, dan $\otimes$ adalah operasi (seperti +, ×, dan operasi lainnya) yang bergantung pada jenis enkripsi yang digunakan [2].

Terdapat dua jenis enkripsi homomorfik, yaitu enkripsi homomorfik sebagian (*partially homomorphic encryption*) dan enkripsi homomorfik penuh (*fully homomorphic encryption*). Enkripsi homomorfik sebagian hanya memungkinkan satu operasi aritmatika pada cipherteks, yaitu antara penjumlahan atau perkalian. Sementara itu, enkripsi homomorfik penuh memungkinkan kedua operasi aritmatika, penjumlahan dan perkalian, pada cipherteks [2].

Pada makalah ini, akan digunakan salah satu algoritme enkripsi homomorfik sebagian, yaitu algoritme Okamoto-Uchiyama.

C. Algoritme Okamoto-Uchiyama

Algoritme Okamoto-Uchiyama merupakan algoritme kunci publik yang dikembangkan oleh Tatsuaki Okamoto dan Shigenori Uchiyama pada tahun 1998. Algoritme ini termasuk dalam *probabilistic encryption*, yang berarti suatu plainteks akan menghasilkan cipherteks yang berbeda setiap kali plainteks tersebut dienkripsi [6]. Selain itu, algoritme ini termasuk ke dalam enkripsi homomorfik sebagian dengan sifat aditif [3].

Proses pembangkitan kunci pada algoritme Okamoto-Uchiyama adalah sebagai berikut.

1. Pilih dua bilangan prima sembarang, p dan q, yang memiliki panjang bit yang sama.
2. Hitunglah $n = p^2 q$.
3. Pilih sembarang bilangan bulat g dalam interval $(1, n-1)$ sedemikian sehingga memenuhi persamaan
$$g^{p-1} \bmod p^2 \neq 1 \quad (3)$$
4. Hitunglah $h = g^n \bmod n$.

Dari proses tersebut akan didapatkan kunci publik (n, g, h) dan kunci privat (p, q).

Proses enkripsi pesan pada algoritme Okamoto-Uchiyama adalah sebagai berikut.

1. Asumsikan m adalah pesan yang ingin dienkripsi yang memenuhi syarat $0 < m < 2^{k-1}$ dengan k adalah panjang bit dari p dan q.
2. Pilih bilangan acak $r \in \mathbb{Z}/n\mathbb{Z}$.
3. Hitung cipherteks c dari plainteks m dengan persamaan
$$c = g^m h^r \bmod n \quad (4)$$

Proses dekripsi cipherteks pada algoritme Okamoto-Uchiyama adalah sebagai berikut.

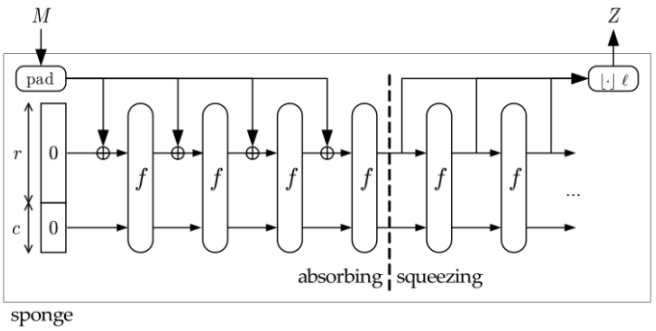
1. Asumsikan c adalah cipherteks yang akan didekripsi.
2. Definisikan fungsi $L(x)$ sebagai
$$L(x) = \frac{x-1}{p} \quad (5)$$
3. Hitunglah a dan b dengan menggunakan persamaan
$$a = c^{p-1} \bmod p^2 \quad (6)$$
$$b = g^{p-1} \bmod p^2 \quad (7)$$
4. Plainteks m dapat dihitung dengan menggunakan persamaan
$$m = \frac{L(a)}{L(b)} \bmod p \quad (8)$$

D. SHA-3

Secure hash algorithm atau yang biasa disingkat dengan SHA merupakan salah satu algoritme fungsi *hash*. Fungsi hash sendiri adalah suatu fungsi yang mengkompresi suatu pesan M berukuran sembarang menjadi suatu *string* h yang bernama *message-digest* atau *hash value* dan berukuran tetap [4].

SHA-3, yang memiliki nama asli Keccak, adalah algoritme fungsi *hash* yang menjadi pemenang dari lomba yang diadakan oleh *National Institute of Standard and Technology* (NIST) untuk membuat sebuah algoritme fungsi *hash* baru. SHA-3 kemudian menjadi komplementer dari fungsi SHA sebelumnya yaitu SHA-1 dan SHA-2.

Dalam membuat *hash value*, SHA-3 menggunakan metode yang disebut dengan konstruksi ‘spons’ (*sponge construction*). Dalam konstruksi ‘spons’ terdapat dua fase, yaitu fase *absorbing* dan fase *squeezing* [5].



Gambar 2 Proses dalam SHA-3 [5]

III. IMPLEMENTASI

A. Skema Basis Data

Pada makalah ini, untuk memudahkan implementasi sistem digunakan sistem basis data yang sederhana. Pada basis data yang digunakan hanya terdapat satu tabel yang digunakan yaitu tabel untuk menyimpan suara yang diberikan oleh.

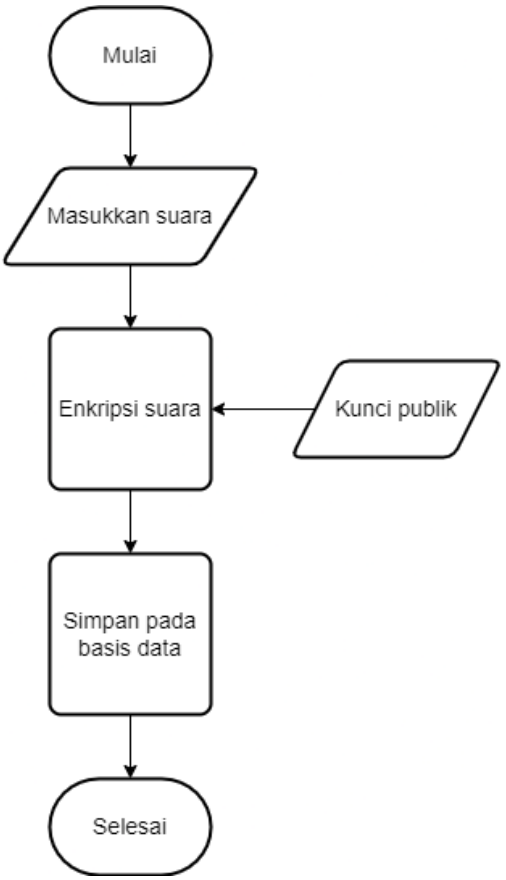
Pada tabel tersebut, informasi yang disimpan adalah informasi mengenai waktu suara diterima oleh sistem, identitas pemilih suara yang merupakan hasil SHA-3 dari nomor induk kependudukan (NIK) pemilih, dan suara (bernilai 0 atau 1) untuk setiap kandidat yang dapat dipilih oleh pemilih.

Votes	
PK	voter_id
	timestamp
	candid_1
	candid_2
	...
	candid_n

Gambar 3 Skema umum basis data yang digunakan

B. Proses Pemberian Suara

Proses pemberian suara dilakukan dengan mengikuti diagram pada gambar 4.



Gambar 4 Tahap-tahap proses pemberian dan penyimpanan suara

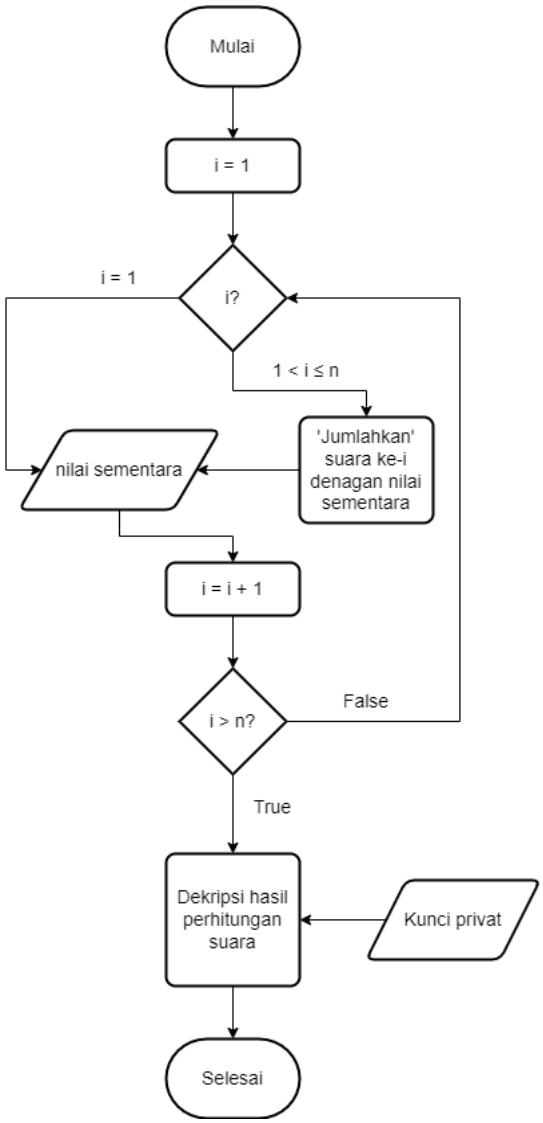
Pertama-tama, sistem akan menerima masukan dari pemilih berupa nomor identitas pemilih dan suara untuk masing-masing kandidat dengan 0 untuk setiap kandidat yang tidak dipilih dan 1 untuk kandidat yang dipilih.

Setelah itu, masukan dari pemilih kemudian akan dienkripsi. Nomor identitas pemilih akan di-'enkripsi' dengan fungsi SHA-3 sementara suara untuk masing-masing kandidat akan dienkripsi dengan algoritme Okamoto-Uchiyama dengan menggunakan kunci publik yang telah dibangkitkan sebelumnya.

Setelah masukan pemilih berhasil dienkripsi, maka hasil enkripsi tersebut akan disimpan pada basis data.

C. Proses Perhitungan Suara

Proses perhitungan suara dilakukan dengan mengikuti diagram pada gambar 5.



Gambar 5 Proses perhitungan suara

Pertama-tama, suara pertama akan dijadikan sebagai hasil sementara. Setelah itu, hasil tersebut akan diperbarui dengan hasil 'penjumlahan' antara nilai sementara dengan suara-suara lainnya dengan menggunakan enkripsi homomorfik aditif sebagaimana persamaan (1).

Ketika semua suara telah digunakan untuk memperbarui nilai sementara, nilai sementara ini kemudian akan didekripsi dengan menggunakan algoritme Okamoto-Uchiyama. Hasil dekripsi ini akan menunjukkan hasil penjumlahan suara untuk masing-masing kandidat.

IV. Hasil Pengujian

Berikut adalah hasil pengujian sederhana pada sistem yang telah dibuat.

A. Pembangkitan Kunci

Untuk membangkitkan kunci yang dapat digunakan pada algoritme Okamoto-Uchiyama, dibutuhkan dua buah bilangan prima yang memiliki panjang bit yang sama. Untuk memudahkan perhitungan yang perlu dilakukan oleh sistem, dipilih dua bilangan prima yang kecil dan memenuhi syarat sebelumnya, yaitu 5519 dan 7013. Dari kedua bilangan tersebut, didapatkan kunci publik dan kunci privat yang dijabarkan pada tabel I sebagai berikut.

TABEL I. Kunci Publik dan Kunci Privat yang Digunakan

Bilangan prima	
p	5519
q	7013
Kunci publik	
n	213611498693
g	13401585519
h	189727378736
Kunci privat	
p	5519
q	7013

B. Pemberian Suara

Dalam pengujian ini, diasumsikan jumlah kandidat yang dapat dipilih oleh seseorang adalah dua orang. Meskipun begitu, sistem yang diimplementasikan masih berlaku untuk jumlah kandidat lainnya. Selain itu, diasumsikan terdapat lima orang pemilih yang masing-masing pemilih tersebut merupakan pemilih yang sudah teregistrasi pada sistem pemilihan. Tabel II berikut menjabarkan daftar pemilih beserta suara yang diberikan untuk masing-masing kandidat.

TABEL II. Informasi Pemilih dan Suara yang Diberikan

No.	NIK	Suara k.1	Suara k.2
1	1111111111111111	0	1
2	2222222222222222	1	0
3	3333333333333333	0	1

4	4444444444444444	1	0
5	5555555555555555	0	1
Total		2	3

Informasi-informasi yang ada pada tabel IV.2 merupakan informasi yang cukup sensitif sehingga perlu di-masking terlebih dahulu sebelum dimasukkan ke dalam basis data. NIK pemilih di-masking dengan menggunakan fungsi SHA-3. Sementara itu, nilai suara untuk masing-masing kandidat perlu di-*masking* dengan menggunakan enkripsi algoritme Okamoto-Uchiyama. Berikut adalah isi dari basis data setelah dilakukan proses *masking*.

TABEL III. ISI BASIS DATA SETELAH MENERIMA SUARA

Suara	Atribut	Nilai
1	*timestamp*	2021-11-27 15:42:00.130523
	identitas	1AFD827639BD0919C0F788EB1C9F80AAABD13AC91949610CBDBBCA909401DD14
	suara untuk kandidat 1	49804642109
	suara untuk kandidat 2	103856844450
2	*timestamp*	2021-11-27 15:42:00.130523
	identitas	C840939042E1596FE83C41E4B10DE197CEB490D4EF13E6F7A9621F25CB327435
	suara untuk kandidat 1	136985125956
	suara untuk kandidat 2	27195361484
3	*timestamp*	2021-11-27 15:42:00.130523
	identitas	057973DEF317E365BF64D675EA635AE5A16410902109E59BE7416022B2033129
	suara untuk kandidat 1	7352868113
	suara untuk kandidat 2	48346303935
4	*timestamp*	2021-11-27 15:42:00.130523
	identitas	4C68EF38B4D3D5697B85C3A696D1C4A1BE5AA6536756754371FA94C0AADFF411
	suara untuk kandidat 1	46338376586
	suara untuk kandidat 2	153206200880
5	*timestamp*	2021-11-27 15:42:00.131523
	identitas	380AF84015F83E74A2F6F63DD2AC97BF927822B4F90668FFC3200EF8740AA5B0
	suara untuk kandidat 1	201397904380
	suara untuk kandidat 2	61728595541

Pada isi basis data di atas, terdapat informasi mengenai waktu suara diterima oleh sistem, NIK yang sudah di-*masking*, dan suara untuk kedua kandidat yang sudah di-*masking*.

C. Penjumlahan Suara

Pada proses penjumlahan suara, suara yang ada pada basis data akan dijumlahkan dengan menggunakan tanpa didekripsi terlebih dahulu dengan memanfaatkan sifat enkripsi homomorfik aditif dari algoritme Okamoto-Uchiyama. Hasil dari proses perhitungan suara untuk masing-masing kandidat ditampilkan pada Tabel IV.

TABEL IV. HASIL PENJUMLAHAN SUARA MASING-MASING KANDIDAT

Jumlah suara kandidat 1	Jumlah suara kandidat 2
109782504189	6270883542

D. Dekripsi Hasil Penjumlahan

Untuk memastikan apakah hasil penjumlahan pada tabel IV sesuai dengan hasil penjumlahan yang sebenarnya yang ada pada tabel II, hasil penjumlahan pada tabel IV akan didekripsi

dengan menggunakan algoritme Okamoto-Uchiyama. Tabel V menampilkan hasil dekripsi penjumlahan suara untuk kedua kandidat.

TABEL V. HASIL DEKRIPSI PENJUMLAHAN SUARA MASING-MASING KANDIDAT

Jumlah suara kandidat 1	Jumlah suara kandidat 2
2	3

Berdasarkan tabel V, maka dapat disimpulkan hasil penjumlahan suara yang dienkripsi dengan menggunakan fungsi adisi sesuai dengan hasil penjumlahan suara secara langsung.

V. KESIMPULAN

Salah satu cara untuk mengantisipasi terjadinya kebocoran informasi penting dan rahasia ketika suatu basis data bocor dan tersebar di khalayak umum adalah dengan menyembunyikan informasi yang ada pada basis data tersebut sehingga orang-orang yang mendapatkan akses basis data tersebut tidak dapat langsung memahami informasi yang ada pada basis data tersebut.

Untuk basis data yang memiliki kardinalitas rendah (jumlah nilai unik yang sedikit) seperti pencatatan suara pada pemilihan umum, salah satu algoritme yang cocok digunakan untuk menyembunyikan data adalah algoritme yang termasuk probabilistic encryption sehingga suatu plainteks dapat menghasilkan cipherteks yang berbeda-beda setiap kali dilakukan enkripsi. Salah satu algoritme yang termasuk dalam probabilistic encryption adalah algoritme kriptografi kunci-publik Okamoto-Uchiyama.

Kelebihan lain dari algoritme Okamoto-Uchiyama adalah bersifat enkripsi homomorfik aditif sehingga proses penjumlahan pada suatu data dapat langsung dilakukan pada cipherteks tanpa harus didekripsi terlebih dahulu.

UCAPAN TERIMA KASIH

Puji syukur ke hadirat Tuhan yang Maha Esa, karena atas berkat dan rahmat-Nya penulis memiliki kesempatan untuk menyelesaikan makalah ini. Selain itu, penulis juga mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T. yang telah memberikan wawasan dan pengetahuan kepada penulis mengenai kriptografi sehingga penulis dapat menyelesaikan penulisan makalah ini.

REFERENSI

[1] R. Munir, "Kriptografi Kunci Publik," 2021, [*Online*], Diakses pada 28 Oktober 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Kriptografi-Kunci-Publik-2020.pdf.

[2] R. Munir, "Enkripsi Homomorfik," 2021, [*Online*], Diakses pada 28 Oktober 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2021-2022/Enkripsi-homomorfik-2021.pdf.

[3] T. Okamoto, S. Uchiyama, A New Public-Key Cryptosystem as Secure as Factoring, 1998, https://doi.org/10.1007/BFb0054135.

[4] R. Munir, "Fungsi Hash," 2021, [*Online*], Diakses pada 09 November 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/Fungsi-hash-2020.pdf.

[5] R. Munir, "SHA-3 (Keccak)," 2021, [*Online*], Diakses pada 09 November 2021, dari https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2020-2021/SHA-3-2020.

[6] Deterministic vs. Probabilistic Encryption, 2016, Diakses pada 27 November 2021 dari https://study.com/academy/lesson/deterministic-vs-probabilistic-encryption.html.

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.

Tangerang Selatan, 04 Desember 2021

Izharulhaq
13518092