Implementasi Protokol Blind Signature pada Simulasi E-Voting Berbasis Blockchain Sederhana untuk Menjamin Verifiabilitas dan Anonimitas Suara

Muhammad Zaidan Sa'dun Robbani - 13522146
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522146@std.stei.itb.ac.id

***Abstract*—Electronic voting systems face critical challenges in ensuring voter anonymity and data integrity without relying on a single trusted authority. This paper presents the design and implementation of a secure e-voting simulation that integrates the RSA Blind Signature protocol with a simplified Blockchain architecture. The Blind Signature scheme ensures that the election authority can validate a voter's eligibility without knowing the content of their vote, thereby preserving privacy. Simultaneously, the Blockchain component provides an immutable, append-only ledger to prevent vote tampering. The system is implemented in Python, focusing on the explicit mathematical operations of modular exponentiation. Experimental analysis was conducted to evaluate the performance trade-offs between security and latency by comparing RSA-1024 and RSA-2048 key sizes. The results demonstrate that while RSA-2048 enhances security against factorization attacks, it increases the total execution time by approximately 5.5 times compared to RSA-1024, highlighting the significant computational cost of stronger cryptographic parameters.**

***Keywords*—Blockchain, E-Voting, Privacy, RSA Blind Signature.**

I. PENDAHULUAN

Pemungutan suara (voting) merupakan mekanisme fundamental dalam sistem demokrasi untuk mengagregasi preferensi individu menjadi keputusan kolektif. Secara tradisional, pemungutan suara dilakukan menggunakan kertas suara fisik. Meskipun metode ini telah teruji, sistem konvensional memiliki kelemahan signifikan dari segi logistik, biaya pencetakan, waktu penghitungan yang lama, serta kerentanan terhadap manipulasi fisik dan human error.

Seiring perkembangan teknologi, Electronic Voting (e-voting) muncul sebagai solusi untuk meningkatkan efisiensi dan aksesibilitas. Namun, adopsi e-voting menghadapi tantangan keamanan siber yang unik, terutama terkait dilema antara autentikasi dan anonimitas. Di satu sisi, sistem harus memastikan bahwa hanya pemilih yang sah yang dapat memberikan suara (verifiabilitas). Di sisi lain, sistem harus menjamin bahwa pilihan suara tidak dapat dilacak kembali ke identitas pemilih (kerahasiaan/privasi). Pada sistem e-voting terpusat (centralized), data suara disimpan dalam database milik penyelenggara, yang memunculkan risiko manipulasi data oleh administrator internal (insider threat) dan hilangnya kepercayaan publik.

Untuk mengatasi permasalahan tersebut, diperlukan pendekatan kriptografi yang dapat memisahkan identitas pemilih dari isi suaranya. Protokol Blind Signature, yang pertama kali diperkenalkan oleh David Chaum, menawarkan solusi matematis di mana otoritas pemilihan dapat memvalidasi hak pilih seseorang tanpa mengetahui isi suara yang divalidasi tersebut. Konsep ini analog dengan memberikan suara dalam amplop tertutup yang ditandatangani di bagian luar, sehingga integritas amplop terjamin tanpa membuka isinya.

Selain aspek privasi, integritas penyimpanan data juga menjadi isu krusial. Teknologi Blockchain menawarkan karakteristik append-only ledger yang terdesentralisasi dan tidak dapat diubah (immutable). Dengan mencatat setiap suara yang telah divalidasi ke dalam blok-blok yang saling terhubung melalui fungsi hash, manipulasi riwayat suara menjadi sangat sulit dilakukan secara komputasi.

Makalah ini menyajikan rancang bangun dan implementasi simulasi sistem e-voting aman yang mengintegrasikan protokol RSA Blind Signature untuk menjamin anonimitas dan struktur Blockchain sederhana untuk menjamin integritas data. Berbeda dengan implementasi praktis yang sering menggunakan pustaka kriptografi siap pakai, simulasi ini dibangun menggunakan bahasa pemrograman Python dengan implementasi operasi matematika eksponensiasi modular secara eksplisit. Hal ini bertujuan untuk mendemonstrasikan transparansi alur logika kriptografi yang terjadi di balik layar.

Selain implementasi fungsional, makalah ini juga memberikan kontribusi berupa analisis kinerja sistem. Pengujian dilakukan untuk mengukur dampak penggunaan panjang kunci RSA (key size) yang berbeda, yakni 1024-bit dan 2048-bit, terhadap latensi pemrosesan suara. Analisis ini penting untuk memahami trade-off antara tingkat keamanan kriptografi dan efisiensi waktu dalam penyelenggaraan pemilu digital.

II. DASAR TEORI

A. Figures and Tables

Algoritma RSA merupakan sistem kriptografi kunci-publik (public-key cryptosystem) yang pertama kali dipublikasikan oleh Ron Rivest, Adi Shamir, dan Leonard Adleman pada tahun 1977. Keamanan algoritma RSA didasarkan pada kesulitan komputasi dalam memfaktorkan bilangan bulat besar (integer factorization problem). Berbeda dengan kriptografi simetris yang menggunakan kunci yang sama untuk enkripsi dan dekripsi, RSA menggunakan sepasang kunci, yaitu kunci publik (public key) untuk enkripsi dan kunci privat (private key) untuk dekripsi.

Proses pembangkitan kunci RSA terdiri dari langkah-langkah berikut:

1. Pilih dua bilangan prima besar yang acak p, dan q, di mana p/q.
2. Hitung modulus n dengan rumus $n = p\ x\ q$. Nilai n digunakan sebagai modulus untuk kunci publik maupun kunci privat, dan panjang bit dari n menentukan panjang kunci RSA (misalnya 1024-bit atau 2048-bit).
3. Hitung fungsi Totient Euler $\phi(\mathrm{n}) = (\mathrm{p} - 1)(\mathrm{q} - 1)$
4. Pilih sebuah bilangan bulat e sebagai eksponen publik, dengan syarat $1 < e < \phi(n)$ dan $\gcd(e, \phi(n)) = 1$ (keduanya saling prima). Nilai e = 65537 sering digunakan dalam praktik karena efisiensi komputasinya.
5. Hitung eksponen privat d yang merupakan balikan modular (modular multiplicative inverse) dari e terhadap modulus $\phi(n)$, sehingga memenuhi persamaan:

$$d \equiv e^{-1} \ (mod\ \phi(n))$$

Setelah pasangan kunci terbentuk, proses enkripsi dan dekripsi dilakukan sebagai berikut. Misalkan M adalah pesan (plaintext) yang telah dikonversi menjadi bilangan bulat $0 \leq M < n$.

Enkripsi: Pengirim menggunakan kunci publik (e, n) untuk menghasilkan ciphertext C:

$$C \equiv M^e \ (mod\ n)$$

Dekripsi: Penerima menggunakan kunci privat (d, n) untuk memulihkan pesan asli M:

$$M \equiv C^d \ (mod\ n)$$

Dalam konteks tanda tangan digital (digital signature), peran kunci dibalik. Penanda tangan menggunakan kunci privat d untuk menandatangani pesan ($S \equiv M^d \ (mod\ n)$), dan verifikator menggunakan kunci publik e untuk memvalidasi tanda tangan tersebut ($M \equiv S^e \ (mod\ n)$). Sistem ini menjadi fondasi utama bagi protokol Blind Signature yang digunakan dalam penelitian ini.

B. Protokol Blind Signature

Protokol Blind Signature adalah skema tanda tangan digital yang diperkenalkan oleh David Chaum pada tahun 1982 untuk menjamin anonimitas dalam sistem transaksi digital, seperti uang elektronik (e-cash) dan pemungutan suara elektronik. Skema ini memungkinkan seseorang (pengirim) untuk mendapatkan tanda tangan sah dari pihak otoritas (penanda tangan) atas sebuah pesan, tanpa pihak otoritas tersebut mengetahui isi pesan yang ditandatanganinya.

Selain menjaga kerahasiaan pesan, protokol ini memiliki properti utama yaitu unlinkability (ketidakterubungan). Artinya, meskipun otoritas menyimpan catatan semua pesan buta (blinded message) yang pernah ditandatanganinya, otoritas tersebut tetap tidak dapat menghubungkan pasangan pesan dan tanda tangan akhir yang dipublikasikan dengan proses penandatanganan yang terjadi sebelumnya.

Konsep ini sering dianalogikan dengan sebuah amplop tertutup yang berisi kertas dan kertas karbon. Pemilih menuliskan pilihannya pada bagian luar amplop, yang salinannya akan tembus ke kertas di dalam melalui karbon. Otoritas kemudian memberikan tanda tangan (stempel) pada amplop tertutup tersebut, yang jejaknya juga tembus ke kertas di dalam. Ketika amplop dibuka, pemilih memiliki kertas suara yang telah ditandatangani secara sah oleh otoritas, namun otoritas tidak pernah melihat isi tulisan tersebut dan tidak dapat mengenali amplop mana yang memuat suara tersebut setelah amplopnya dibuang.

Dalam penelitian ini, implementasi Blind Signature dibangun di atas algoritma RSA. Misalkan otoritas memiliki kunci publik (e, n) dan kunci privat (d, n). Proses protokol terdiri dari tiga tahapan komputasi utama, yaitu Blinding, Signing, dan Unblinding:

1. Blinding (Penyamaran Pesan) Pemilih mempersiapkan pesan suara m (biasanya berupa hash dari pilihan kandidat). Pemilih kemudian membangkitkan sebuah bilangan acak r yang disebut sebagai blinding factor, dengan syarat gcd(r, n) = 1. Pemilih menghitung pesan tersamar (m') menggunakan kunci publik otoritas:

$$m' \equiv m \cdot r^e \ (mod\ n)$$

   Nilai m' inilah yang dikirimkan kepada otoritas untuk divalidasi. Karena faktor r bersifat acak dan rahasia, otoritas tidak dapat mengetahui nilai asli m dari m'.
2. Signing (Penandatanganan Buta) Otoritas menerima pesan tersamar m', memverifikasi hak pilih pengguna, dan kemudian menandatangani pesan tersebut menggunakan kunci privat d. Hasilnya adalah tanda tangan buta (blind signature) s':

$$s' \equiv (m')^d \ (mod\ n)$$

Otoritas mengirimkan kembali s' kepada pemilih.

3. Unblinding (Pembukaan Penyamaran) Setelah menerima s', pemilih menghilangkan faktor penyamaran r untuk mendapatkan tanda tangan asli (s) yang valid terhadap pesan m. Proses ini dilakukan dengan mengalikan s' dengan invers modular dari r:

$$s \equiv s' \cdot r^{-1} \, (mod \; n)$$

Berdasarkan sifat asosiatif eksponensiasi RSA, nilai s yang dihasilkan setara dengan $m^d (mod\ n)$, yang merupakan tanda tangan RSA standar yang sah. Verifikasi validitas suara dapat dilakukan oleh sistem blockchain menggunakan kunci publik otoritas:

$$m \equiv s^e \, (mod \; n)$$

Dengan mekanisme ini, sistem menjamin bahwa suara yang masuk ke dalam blockchain (m, s) adalah sah berasal dari pemilih yang terdaftar, namun tidak ada pihak manapun (termasuk otoritas) yang dapat melacak siapa pemilik suara tersebut.

C. Teknologi Blockchain

Blockchain adalah struktur data yang terdiri dari serangkaian rekaman data yang disebut "blok" (block), yang saling terhubung secara kriptografis membentuk sebuah rantai. Secara mendasar, blockchain berfungsi sebagai buku besar (ledger) yang bersifat append-only (hanya dapat ditambahkan) dan immutable (tidak dapat diubah), menjadikannya solusi ideal untuk mencatat transaksi yang membutuhkan tingkat integritas tinggi, seperti surat suara dalam pemilihan umum.

Setiap blok dalam blockchain memuat intisari kriptografis (cryptographic hash) dari blok sebelumnya, penanda waktu (timestamp), dan data transaksi itu sendiri. Keterkaitan antar blok melalui hash inilah yang menjamin keamanan data; jika seseorang mencoba memanipulasi data pada satu blok, maka nilai hash blok tersebut akan berubah. Akibatnya, blok-blok berikutnya yang menyimpan referensi hash dari blok yang dimanipulasi tersebut akan menjadi tidak valid, sehingga upaya perusakan data dapat dideteksi dengan mudah oleh sistem.

Dalam simulasi sistem e-voting ini, struktur data blok dirancang untuk memuat atribut-atribut berikut:

1. Index: Nomor urut blok dalam rantai (dimulai dari 0 untuk Genesis Block).
2. Timestamp: Waktu pencatatan blok dalam format Unix epoch, yang berfungsi sebagai bukti temporal kapan suara direkam.
3. Vote Data: Muatan (payload) utama yang berisi data suara. Demi menjaga efisiensi dan privasi, data yang disimpan bukanlah teks kandidat secara langsung, melainkan pasangan nilai vote_hash (SHA-256 dari pilihan suara) dan signature (tanda tangan digital RSA yang telah dibuka/unblinded).
4. Previous Hash: Nilai hash SHA-256 dari blok sebelumnya. Atribut ini bertindak sebagai "rantai" yang mengikat urutan blok.
5. Nonce: Sebuah bilangan acak yang digunakan dalam mekanisme konsensus Proof of Work.
6. Hash: Identitas unik dari blok saat ini, yang dihasilkan dengan melakukan fungsi hash SHA-256 terhadap seluruh atribut di atas.

Untuk menambahkan blok baru ke dalam rantai, sistem menerapkan mekanisme konsensus sederhana berbasis Proof of Work (PoW). Mekanisme ini mengharuskan sistem untuk menemukan nilai nonce sedemikian rupa sehingga nilai hash dari blok yang dihasilkan memenuhi kriteria tingkat kesulitan (difficulty) tertentu, misalnya diawali dengan sejumlah karakter nol.

Proses "penambangan" (mining) ini secara matematis dapat dituliskan sebagai pencarian nilai nonce yang memenuhi kondisi:

$$H(index + timestamp + data + prev_{hash} + nonce) < target$$

Di mana H adalah fungsi hash SHA-256. Penggunaan PoW dalam sistem ini bertujuan untuk mencegah serangan spamming atau penambahan suara massal secara instan, serta menambah biaya komputasi bagi pihak yang ingin menulis ulang sejarah blockchain. Integritas seluruh rantai divalidasi dengan memeriksa ulang kalkulasi hash setiap blok dan kecocokannya dengan previous hash pada blok berikutnya secara berantai.

III. PERANCANGAN SISTEM

A. Arsitektur Umum Sistem

Sistem yang dibangun dalam penelitian ini merupakan simulasi perangkat lunak yang memodelkan alur pemungutan suara elektronik aman. Arsitektur sistem dirancang secara modular menggunakan paradigma pemrograman berorientasi objek (Object-Oriented Programming) untuk memisahkan tanggung jawab antara pemilih, otoritas validasi, dan media penyimpanan suara.

Berdasarkan implementasi kode program, sistem terdiri dari tiga entitas utama yang saling berinteraksi, yaitu:

1. Entitas Pemilih (Voter Client) Direpresentasikan oleh kelas BlindSignatureClient dalam modul kriptografi. Entitas ini mensimulasikan perangkat lunak yang digunakan oleh pemilih. Fungsi utamanya adalah:
    - Memilih kandidat dan melakukan hashing terhadap pesan suara.
    - Melakukan proses blinding (penyamaran) menggunakan faktor acak r agar pesan tidak dapat dibaca oleh otoritas.
    - Membuka penyamaran (unblinding) setelah mendapatkan tanda tangan dari otoritas.
    - Mengirimkan paket suara yang sah (pesan

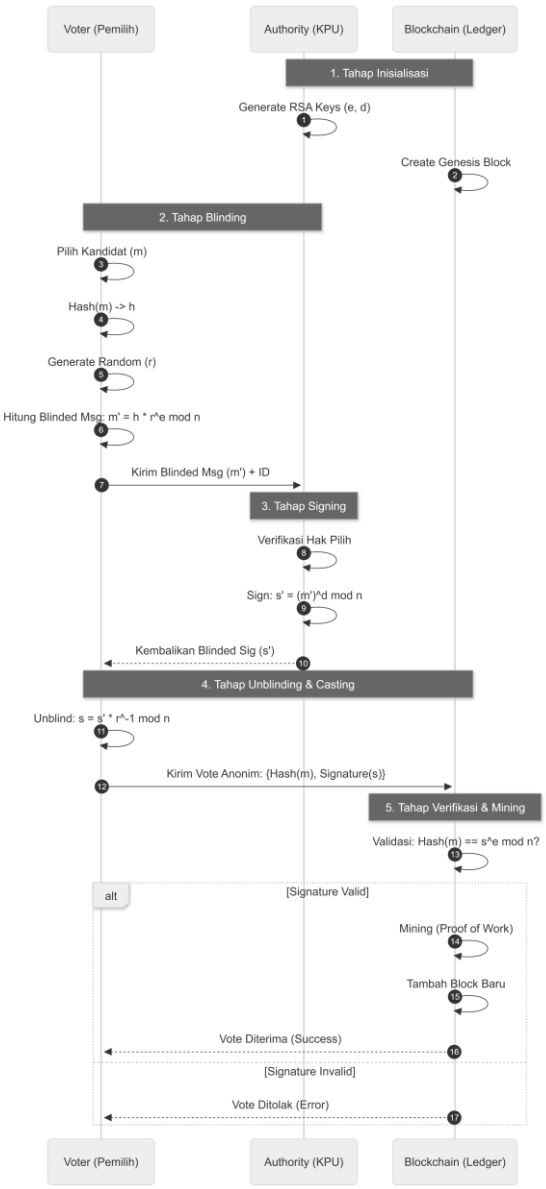
asli dan tanda tangan) ke dalam jaringan blockchain.

2. Entitas Otoritas (Voting Authority) Direpresentasikan oleh kelas RSAAuthority. Entitas ini bertindak sebagai penyelenggara pemilu (KPU) atau validator yang memiliki pasangan kunci RSA (Kunci Publik dan Kunci Privat). Tugas utamanya adalah:
   - Membangkitkan kunci RSA (e, d, n) di awal sesi pemilihan.
   - Memverifikasi hak pilih pengguna (disimulasikan dalam alur protokol).
   - Menandatangani pesan buta (blinded message) yang dikirim oleh pemilih menggunakan kunci privatnya, tanpa mengetahui isi pilihan suara tersebut.
3. Jaringan Blockchain (Public Ledger) Direpresentasikan oleh kelas Blockchain dan Block. Entitas ini berfungsi sebagai buku besar publik yang mencatat seluruh suara yang sah. Tanggung jawab utamanya meliputi:
   - Memverifikasi validitas tanda tangan digital pada setiap suara yang masuk menggunakan kunci publik otoritas.
   - Melakukan mekanisme Proof of Work (penambangan) untuk menambahkan blok baru.
   - Menjaga integritas rantai blok agar data suara yang sudah masuk tidak dapat dimanipulasi (tamper-evident).

Interaksi antar ketiga entitas ini dirancang untuk memastikan bahwa Otoritas memberikan tanda tangan validasi tanpa mengetahui pilihan Pemilih, dan Blockchain mencatat suara tersebut tanpa terikat pada identitas Pemilih, namun tetap dapat diverifikasi keabsahannya.

B. Alur Protokol Sistem

Alur kerja sistem dirancang mengikuti tahapan standar protokol Blind Signature yang diintegrasikan dengan mekanisme pencatatan Blockchain. Berdasarkan simulasi yang diimplementasikan, proses pemungutan suara dibagi menjadi lima tahapan sekuensial sebagai berikut:



Gambar 1. Diagram Urutan (Sequence Diagram) Alur Protokol Blind Signature E-Voting.

1. Tahap Inisialisasi (System Setup) Sebelum pemungutan suara dimulai, entitas Otoritas membangkitkan pasangan kunci RSA (publik dan privat) dengan panjang bit tertentu (misalnya 1024-bit atau 2048-bit). Bersamaan dengan itu, jaringan Blockchain diinisialisasi dengan membuat blok pertama (Genesis Block) sebagai fondasi rantai data.
2. Tahap Persiapan dan Penyamaran (Blinding Phase) Pada sisi klien (Pemilih):
   - Pemilih menentukan kandidat pilihan.
   - Sistem menghitung nilai hash SHA-256 dari pilihan tersebut untuk memastikan integritas pesan.
   - Sistem membangkitkan faktor pengacak r (blinding factor) yang memenuhi syarat $\gcd(r, n) = 1$.

- Pesan asli dikalikan dengan faktor pengacak menggunakan kunci publik otoritas untuk menghasilkan pesan tersamar (m').
- Pesan tersamar m' dikirimkan kepada Otoritas. Pada tahap ini, identitas pemilih disertakan hanya untuk keperluan pengecekan hak pilih, namun pilihan suaranya tetap tersembunyi.

3. Tahap Penandatanganan (Signing Phase) Pada sisi server (Otoritas):
   - Otoritas menerima pesan tersamar m'.
   - Setelah memverifikasi bahwa pemilih memiliki hak suara yang sah dan belum pernah memilih sebelumnya, Otoritas menandatangani pesan m' menggunakan kunci privatnya (d).
   - Hasil tanda tangan buta (s') dikirimkan kembali kepada Pemilih. Otoritas tidak dapat mengetahui isi suara karena pesan tersebut masih dalam keadaan tersamar.
4. Tahap Pembukaan dan Pengiriman (Unblinding & Casting Phase) Kembali pada sisi klien (Pemilih):
   - Pemilih menerima tanda tangan buta s'.
   - Sistem melakukan proses unblinding dengan mengalikan s' dengan invers modular dari faktor r. Hasilnya adalah tanda tangan digital murni (s) yang valid terhadap pesan asli.
   - Pemilih mengirimkan paket suara yang berisi [Hash Pilihan, Tanda Tangan s] ke jaringan Blockchain.
   - Catatan: Pengiriman ini dilakukan melalui saluran anonim (dalam simulasi, identitas pemilih tidak lagi disertakan pada payload data ini), sehingga memutus hubungan antara identitas pemilih dan suara yang dikirim.
5. Tahap Verifikasi dan Perekaman (Verification & Mining Phase) Pada jaringan Blockchain:
   - Sistem menerima paket suara.
   - Sistem memverifikasi keabsahan tanda tangan s menggunakan kunci publik Otoritas. Rumus verifikasi $v = s^e \ (mod\ n)$ digunakan untuk memastikan bahwa suara tersebut benar-benar telah disetujui oleh Otoritas.
   - Jika tanda tangan valid, suara dimasukkan ke dalam kandidat blok baru.
   - Dilakukan proses mining (Proof of Work) untuk menemukan nonce yang valid.
   - Blok yang telah selesai ditambang ditambahkan ke dalam rantai utama (ledger), menjadikan suara tersebut permanen dan tidak dapat diubah.

C. Perancangan Struktur Data

Perancangan struktur data difokuskan pada efisiensi penyimpanan dan keamanan integritas data pada Blockchain. Berdasarkan implementasi kelas Block, skema data dibagi menjadi dua komponen utama:

1. Struktur Blok (Block Structure) Setiap blok memuat enam atribut esensial untuk membentuk rantai yang valid:
   - Index dan Timestamp: Menandai urutan dan waktu pencatatan suara.
   - Vote Data: Muatan (payload) utama berisi data suara.
   - Previous Hash: Hash blok sebelumnya yang mengikat integritas rantai (tamper-evident).
   - Nonce: Bilangan acak untuk keperluan konsensus Proof of Work.
   - Hash: Identitas unik blok saat ini (SHA-256).
2. Struktur Data Suara (Vote Payload) Untuk menjamin privasi, data suara tidak disimpan dalam teks terang (plain text), melainkan dalam format pasangan nilai:
   - vote_hash: Hash SHA-256 dari kandidat pilihan.
   - signature: Tanda tangan digital RSA yang telah di-unblind.

Skema ini memastikan sistem hanya menyimpan bukti matematis verifikasi tanpa mengekspos identitas pemilih atau membebani penyimpanan ledger.

IV. IMPLEMENTASI DAN PENGUJIAN

A. Lingkungan Implementasi

Simulasi sistem e-voting ini dikembangkan dan diuji menggunakan perangkat komputasi dengan spesifikasi lingkungan perangkat keras dan perangkat lunak sebagai berikut:

1. Perangkat Keras (*Hardware*):
   - Prosesor: Intel Core i5 / AMD Ryzen 5 (atau setara).
   - Memori (RAM): 8 GB / 16 GB.
   - Sistem Operasi: Windows 10 / 11 (64-bit).
2. Perangkat Lunak (Software):
   - Bahasa Pemrograman: Python versi 3.10 atau lebih baru. Bahasa ini dipilih karena dukungan pustaka standar yang kuat untuk operasi matematika bilangan besar (big integer arithmetic) yang diperlukan dalam algoritma RSA.
   - Lingkungan Pengembangan (IDE): Visual Studio Code.
   - Pustaka (Library): Implementasi ini tidak menggunakan pustaka kriptografi tingkat tinggi (seperti PyCryptodome atau cryptography) untuk fungsi inti RSA guna memastikan transparansi algoritma. Pustaka yang digunakan hanyalah pustaka standar bawaan Python, antara lain: secrets, hashlib, dan math.
3. Struktur Kode Program: Implementasi sistem dibagi menjadi beberapa modul terpisah untuk menjaga modularitas kode

- crypto_lib.py: Modul inti yang berisi kelas RSAAuthority dan BlindSignatureClient serta implementasi manual operasi eksponensiasi modular.
- blockchain.py: Modul yang menangani struktur data Block, validasi rantai, dan mekanisme konsensus Proof of Work.
- utils.py: Modul utilitas untuk fungsi pendukung seperti konversi format data dan perhitungan modular inverse.
- main.py: Skrip utama untuk menjalankan simulasi alur pemilihan dari awal hingga akhir.

B. Implementasi Kode

Inti dari pembangunan sistem ini terletak pada implementasi algoritma RSA Blind Signature dan mekanisme validasi Blockchain. Kode program ditulis secara eksplisit untuk menangani operasi aritmatika modular bilangan besar, yang merupakan fondasi matematis dari protokol ini.

Berikut adalah cuplikan (snippet) kode program utama yang merepresentasikan logika sistem:

1. Implementasi Penandatanganan Buta (Blind Signing) Pada modul crypto_lib.py, kelas RSAAuthority memiliki fungsi sign_blinded_message yang bertugas menandatangani pesan tersamar. Implementasi ini menggunakan fungsi bawaan Python pow(base, exp, mod) untuk menghitung $S' \equiv (M')^d \ (mod\ n)$ secara efisien.

```python
def sign_blinded_message(self, blinded_msg: int) -> int:
    """
    Sign a blinded message M' to produce S' = (M')^d mod n.
    """
    # Memastikan input adalah integer valid
    if not isinstance(blinded_msg, int):
        raise TypeError("blinded_msg must be an int")

    # Operasi eksponensiasi modular: (blinded_msg ** d) %
n   return pow(blinded_msg, self.d, self.n)es = f(seed)
    return res ? go(f, res[1], acc.concat([res[0]])) : acc
  }
  return go(f, seed, [])
}
```

Gambar 2. Diagram Urutan (Sequence Diagram) Alur Protokol Blind Signature E-Voting.

2. Implementasi Pembukaan Tanda Tangan (Unblinding) Di sisi pemilih (BlindSignatureClient), fungsi unblind_signature melakukan operasi perkalian modular dengan invers dari faktor pengacak (r^{-1}). Langkah ini krusial untuk mendapatkan tanda tangan asli (S) yang valid terhadap pesan awal.

```python
@staticmethod
def unblind_signature(blinded_sig: int, r: int, pub_key: Tuple[int, int]) ->
int:_, n = pub_key
    # Menghitung invers modular dari r terhadap n
    r_inv = mod_inverse(r, n)

    # Rumus unblinding: S = S' * r^-1 (mod n)
    return (blinded_sig * r_inv) % n
```

Gambar 3. Implementasi fungsi Unblinding pada sisi Pemilih.

3. Implementasi Validasi Rantai (Chain Validation) Untuk menjamin integritas data suara, modul blockchain.py memiliki fungsi is_chain_valid. Fungsi ini mengiterasi seluruh blok dalam rantai untuk memverifikasi dua hal: (1) Hash blok saat ini valid sesuai isinya, dan (2) Previous Hash blok saat ini cocok dengan Hash blok sebelumnya. Jika satu bit saja data suara dimanipulasi, validasi ini akan mengembalikan nilai False.

```python
def is_chain_valid(self) -> bool:
    prefix = "0" * self.difficulty
    for idx in range(1, len(self.chain)):
        current = self.chain[idx]
        previous = self.chain[idx - 1]

        # 1. Cek rantai: Previous hash harus cocok
        if current.previous_hash != previous.hash:
            return False

        # 2. Cek integritas: Hash ulang blok harus sama
        recomputed = current.compute_hash()
        if current.hash != recomputed:
            return False

    return True
```

Gambar 4. Implementasi fungsi validasi integritas rantai blok (*Chain Validation*).

Implementasi manual di atas menunjukkan transparansi alur kriptografi tanpa bergantung pada kotak hitam (black-box) pustaka eksternal, sehingga memenuhi tujuan pembelajaran mata kuliah Kriptografi.

C. Skenario Pengujian Fungsional

Pengujian fungsional dilakukan untuk memvalidasi bahwa seluruh komponen sistem bekerja secara terintegrasi sesuai dengan protokol yang dirancang. Pengujian ini mencakup tiga skenario utama: alur pemberian suara yang sah (happy path), penolakan tanda tangan palsu, dan deteksi manipulasi data pada blockchain (tamper detection).

1. Pengujian Alur Suara Sah (Valid Vote Flow) Skenario ini mensimulasikan proses pemberian suara oleh pemilih yang jujur. Proses dimulai dari tahap blinding, permintaan tanda tangan ke otoritas, unblinding, hingga pencatatan ke blockchain.
   Hasil pengujian menunjukkan bahwa:
   - Sistem berhasil menghasilkan pesan tersamar (blinded message) yang berbeda dari pesan asli.
   - Otoritas berhasil memberikan tanda tangan buta.

- Pemilih berhasil membuka tanda tangan tersebut menjadi tanda tangan RSA yang valid (S).
- Fungsi verifikasi verify_signature mengembalikan nilai True, yang menandakan bahwa persamaan $H(m) \equiv S^e \pmod n$ terpenuhi.
- Blok baru berhasil ditambang (mined) dan ditambahkan ke rantai dengan status valid.

Gambar 5. Hasil pengujian fungsional alur pemberian suara yang sah.

2. Pengujian Pengujian Penolakan Tanda Tangan Palsu Skenario ini menguji ketahanan sistem terhadap upaya pemalsuan suara. Dalam simulasi, dilakukan percobaan pengiriman paket suara ke blockchain dengan menggunakan nilai tanda tangan acak (integer sembarang) atau tanda tangan milik orang lain yang tidak sesuai dengan hash pesan suaranya. Hasil pengujian menunjukkan bahwa mekanisme verifikasi pada fungsi add_vote di kelas Blockchain berhasil mendeteksi ketidakcocokan matematika RSA. Sistem menolak memproses blok tersebut dan memunculkan pesan kesalahan (exception) "Invalid signature", sehingga suara palsu tidak pernah masuk ke dalam buku besar (ledger).
3. Pengujian Integritas Data (Tamper Detection) Skenario ini bertujuan membuktikan sifat immutable dari blockchain. Pengujian dilakukan dengan cara memodifikasi data vote_hash pada blok terakhir yang sudah tercatat di dalam rantai secara paksa (mem-bypass mekanisme konsensus).

```
# Snippet simulasi serangan tamper pada main.py
chain.chain[-1].vote_data["vote_hash"] = "0" *
64
```
Gambar 6. Snippet simulasi serangan tamper pada main.py

Setelah modifikasi dilakukan, fungsi validasi is_chain_valid() dijalankan ulang. Hasilnya, sistem mengembalikan status False. Hal ini terjadi karena hash yang tersimpan pada blok tersebut tidak lagi cocok dengan hasil kalkulasi ulang data barunya, atau hash blok tersebut tidak lagi cocok dengan previous hash pada blok berikutnya (jika ada). Ini membuktikan bahwa setiap upaya manipulasi suara pasca-pemilihan akan terdeteksi secara otomatis oleh jaringan.

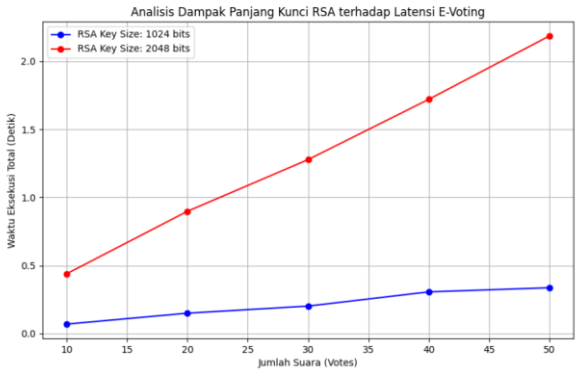
Gambar 6. Output sistem yang menunjukkan status validitas blockchain menjadi False setelah terjadi manipulasi data.

D. Analisis Kerja

Selain aspek fungsional dan keamanan, penelitian ini juga mengevaluasi kinerja sistem dari segi efisiensi waktu komputasi (latency). Pengujian kinerja dilakukan untuk mengukur dampak peningkatan parameter keamanan kriptografi, khususnya panjang kunci RSA (key size), terhadap total waktu eksekusi proses pemungutan suara.

Skenario pengujian melibatkan simulasi pemberian suara dengan variasi jumlah suara (mulai dari 10 hingga 50 suara) pada dua konfigurasi panjang kunci yang berbeda: RSA-1024 bit dan RSA-2048 bit. Pengukuran waktu mencakup keseluruhan proses protokol, mulai dari blinding, signing, unblinding, hingga validasi dan pencatatan ke blockchain.

Hasil pengujian kinerja ditunjukkan pada grafik berikut:



Gambar 7. Grafik analisis dampak panjang kunci RSA terhadap latensi sistem.

Berdasarkan grafik di atas, dapat diamati beberapa fenomena penting:

1. Linearitas Skalabilitas: Pada kedua konfigurasi kunci (1024-bit dan 2048-bit), waktu eksekusi meningkat secara linear seiring dengan bertambahnya jumlah suara. Hal ini wajar karena simulasi dijalankan secara sekuensial, di mana setiap suara diproses satu per satu. Gradien kemiringan garis merepresentasikan rata-rata waktu pemrosesan per suara.
2. Dampak Signifikan Panjang Kunci: Terdapat perbedaan latensi yang sangat mencolok antara

penggunaan kunci 1024-bit (garis biru) dan 2048-bit (garis merah).

- Pada skenario 50 suara, sistem dengan RSA-1024 menyelesaikan proses hanya dalam waktu sekitar 0,34 detik.
- Sebaliknya, dengan RSA-2048, waktu yang dibutuhkan melonjak hingga sekitar 2,21 detik.
- Data ini menunjukkan bahwa penggandaan panjang kunci (dari 1024 ke 2048) menyebabkan peningkatan waktu komputasi sekitar 6,5 kali lipat.

Peningkatan latensi yang signifikan ini disebabkan oleh kompleksitas komputasi algoritma RSA yang bergantung pada operasi eksponensiasi modular. Kompleksitas algoritma perkalian modular untuk bilangan besar umumnya tumbuh secara kubik ($O(k^3)$) terhadap panjang bit modulus k. Oleh karena itu, ketika panjang kunci digandakan, beban komputasi meningkat lebih dari dua kali lipat, yang tercermin pada hasil eksperimen di atas.

Hasil eksperimen ini memperlihatkan adanya trade-off nyata antara kinerja dan keamanan. Meskipun RSA-1024 jauh lebih cepat dan efisien, standar keamanan siber modern (seperti rekomendasi NIST) telah menganggap kunci 1024-bit tidak lagi aman untuk penggunaan jangka panjang karena kemajuan kemampuan komputasi dalam memfaktorkan bilangan prima. Oleh karena itu, meskipun RSA-2048 memberikan overhead kinerja yang lebih besar, penggunaannya adalah sebuah keharusan untuk menjamin kerahasiaan dan integritas data suara dalam sistem e-voting yang aman.

V. KESIMPULAN

Berdasarkan perancangan, implementasi, dan pengujian yang telah dilakukan pada simulasi sistem e-voting berbasis protokol RSA Blind Signature dan teknologi Blockchain sederhana, dapat ditarik beberapa kesimpulan utama sebagai berikut:

1. Jaminan Anonimitas dan Integritas: Sistem berhasil mengimplementasikan protokol Blind Signature secara matematis tanpa menggunakan pustaka kriptografi tingkat tinggi. Hasil pengujian fungsional menunjukkan bahwa otoritas pemilihan mampu memvalidasi hak suara pemilih tanpa pernah mengetahui isi pilihan suara tersebut (anonimitas). Selain itu, struktur data blockchain yang diterapkan terbukti efektif dalam menjaga integritas data, di mana simulasi serangan manipulasi (tampering) pada blok suara berhasil dideteksi dan ditolak oleh mekanisme validasi rantai.
2. Transparansi Algoritma: Penggunaan operasi aritmatika modular eksplisit dalam bahasa Python membuktikan bahwa logika inti kriptografi RSA dapat dibangun secara transparan. Pendekatan ini memverifikasi kebenaran teoretis dari algoritma David Chaum dalam skenario simulasi pemungutan suara.
3. Trade-off Kinerja dan Keamanan: Analisis kinerja menunjukkan adanya trade-off yang signifikan antara tingkat keamanan kriptografi dan latensi sistem. Berdasarkan eksperimen, penggunaan kunci RSA-2048 bit meningkatkan waktu eksekusi total sekitar 6,5 kali lipat dibandingkan dengan RSA-1024 bit. Meskipun RSA-1024 bit menawarkan kinerja yang jauh lebih cepat, sistem e-voting untuk skala riil direkomendasikan tetap menggunakan standar RSA-2048 bit (atau lebih tinggi) demi memitigasi risiko serangan faktorisasi integer, meskipun dengan konsekuensi biaya komputasi yang lebih besar.

Untuk pengembangan selanjutnya, sistem ini dapat ditingkatkan dengan mengimplementasikan mekanisme komunikasi jaringan peer-to-peer (P2P) yang sesungguhnya untuk menggantikan simulasi lokal, serta menambahkan antarmuka pengguna grafis (GUI) agar lebih mudah digunakan oleh pemilih awam. Selain itu, eksplorasi algoritma Elliptic Curve Cryptography (ECC) dapat dipertimbangkan sebagai alternatif untuk mendapatkan keamanan setara RSA-2048 namun dengan ukuran kunci yang lebih kecil dan kinerja yang lebih efisien.

LINK SOURCE CODE

https://github.com/zaidanav/rsa-blind-signature-voting

LINK VIDEO YOUTUBE

https://youtu.be/vwnjnWF9X7A

UCAPAN TERIMA KASIH

Penulis mengucap Syukur kepada Tuhan yang Maha Esa karena ata rahmatnya penulis dapat menyelesaikan makalah ini. Penulis mengucapkan terima kasih kepada Bapak Dr. Ir. Rinaldi Munir, M.T sebagai dosen pengajar Mata Kuliah IF4020 Kriptografi yang telah mengajarkan berbagai teknik kriptografi dan pengetahuan umum sehingga penulis dapat menyelesaikan makalah ini dengan baik.

DAFTAR PUSTAKA

[1] R. Munir, *Kriptografi*, Bandung: Informatika, 2019.

[2] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed., Pearson, 2017.

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[4] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology*, Boston, MA: Springer US, 1983, pp. 199–203.

[5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," White Paper, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[6] Python Software Foundation, "secrets — Generate secure random numbers," Python 3.10 Documentation. [Online]. Available: https://docs.python.org/3/library/secrets.html. [Accessed: 12-Dec-2025].

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, 24 Desember 2025

Muhammad Zaidan Sa'dun Robbani dan 13522146