Implementasi Enkripsi Homomorfik dalam Kerahasiaan Data Donasi

Damianus Clairvoyance Diva Putra - 13520035
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: 13520035@std.stei.itb.ac.id

***Abstract*—Indonesia menempati peringkat 1 sebagai negara paling dermawan di dunia, dengan donasi secara daring sebagai pilihan yang lebih populer dibandingkan donasi secara luring. Untuk mendukung hal tersebut, platform donasi daring perlu mengutamakan etika, akuntabilitas, dan tranparansi dalam proses bisnisnya, salah satunya ialah dengan tidak menyimpan nominal donasi setiap donatur sebagai plainteks, tetapi tetap memastikan seluruh dana terkumpul dengan sesuai. Enkripsi homomorfik dapat menjawab kebutuhan itu, yaitu dengan menjumlahkan nominal donasi dalam bentuk cipherteks saja, tetapi tetap menghasilkan nilai yang sama dengan penjumlahan dalam bentuk plainteks.**

Keywords—enkripsi homomorfik, kerahasiaan, donasi

I. Latar Belakang

Menurut laporan World Giving Index 2023 yang dirilis oleh Charities Aid Foundation (CAF), sebuah badan amal asal Inggris, Indonesia menempati peringkat 1 sebagai negara paling dermawan di dunia. Adapun indikator perilaku yang menjadi penilaian utama salah satunya termasuk donasi uang untuk amal [1]. Selain itu, berdasarkan tSurvey.id, sebuah survei digital oleh Telkomsel, mayoritas responden (warga Indonesia) berdonasi secara daring lebih dari satu kali dalam sebulan (59%), serta frekuensi donasinya secara daring lebih banyak daripada secara luring (50%) [2].

Untuk memenuhi kebutuhan masyarakat dalam berdonasi tersebut, platform donasi daring harus melaksanakan praktik terbaik (*best practices*) sebagai organisasi nirlaba (*non-profit*). National Council of Nonprofits menyebutkan salah tiganya, yaitu akuntabilitas, transparansi, dan etika [3]. Akuntabilitas berarti organisasi memiliki laporan yang teratur terkait pengumpulan dan penggunaan dana, sementara transparansi berarti organisasi membuka akses laporan tersebut kepada para donatur. Terakhir, salah satu perwujudan etika ialah dengan menghargai privasi donatur, yaitu dengan memberikan opsi kepada donatur untuk merahasiakan atau mengumumkan nominal donasi mereka kepada publik. Namun, donatur masih harus dapat melihat nominal donasinya sendiri.

II. Dasar Teori

A. Kriptografi Kunci Publik

Kriptografi kunci publik (*public key cryptography*), atau disebut juga kriptografi kunci nir-simetri (*assymetric key cryptography*), menjawab masalah kriptografi kunci simetri (*symmetric key cryptography*), yaitu bahwa pengiriman kunci rahasia yang sama antara dua pihak memerlukan saluran yang aman, tetapi tetap cepat dan murah. Dalam kriptografi kunci publik, kedua pihak tidak perlu menyepakati kunci rahasia yang sama, tetapi setiap pihak dapat memiliki sepasang kuncinya sendiri, yaitu kunci publik (*public key*; PK) dan kunci privat (*private*/*secret key*; SK). Dengan begitu, pengiriman kunci rahasia tidak diperlukan.

Misalkan saja Alice ingin mengirim pesan kepada Bob, maka Alice akan mengenkripsi pesannya dengan kunci publik Bob, lalu mengirimkan hasil enkripsi tersebut kepada Bob, baru kemudian Bob akan mendekripsi cipherteks tersebut dengan kunci privat dirinya. Mekanisme ini berlaku pula sebaliknya, yaitu jika Bob ingin mengirim pesan kepada Alice. Perhatikan bahwa tidak hanya pihak terpilih yang dapat berkomunikasi secara aman dengan kita, tetapi seluruh pihak yang mengetahui kunci publik kita dapat mengirim pesan kepada kita secara rahasia. Metode ini pun dinilai aman karena secara komputasi hampir tidak mungkin bagi pihak ketiga untuk menurunkan kunci privat dari kunci publik.

$$E_{PK1}(P) = C \quad (1)$$
$$D_{SK1}(C) = P \quad (2)$$

dengan E adalah fungsi enkripsi, D adalah fungsi dekripsi, $PK1$ adalah kunci publik pihak ke-1, $SK1$ adalah kunci privat pihak ke-1, P adalah plainteks, dan C adalah cipherteks.

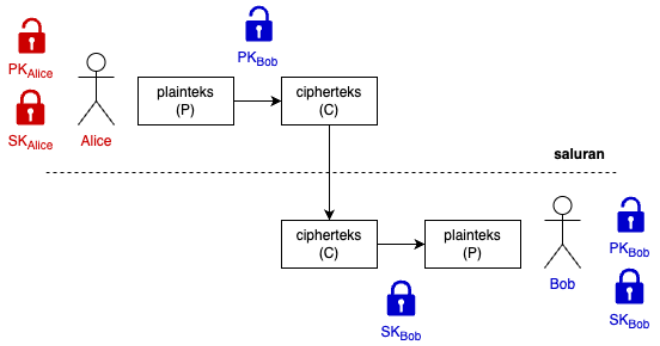

Fig. 1. Kriptografi Kunci Publik

B. Enkripsi Homomorfik

Enkripsi homomorfik menjawab masalah metode enkripsi konvensional, yaitu bahwa manipulasi cipherteks, misalnya operasi matematik (penjumlahan) atau statistik (perhitungan rerata) sederhana menyisakan suatu celah keamanan dalam prosesnya. Spesifiknya, cipherteks perlu didekripsi terlebih dahulu menjadi plainteks, dimanipulasi sesuai kebutuhan, baru kemudian dienkripsi kembali menjadi cipherteks. Proses manipulasi inilah yang rentan terhadap serangan pihak ketiga. Dengan metode enkripsi homomorfik, komputasi pada cipherteks dapat dieksekusi tanpa perlu didekripsi terlebih dahulu, tetapi dijamin bahwa hasilnya akan sama dengan komputasi pada plainteks. Secara formal, misalkan E menyatakan fungsi enkripsi. E dikatakan homomorfik, yaitu jika diberikan $E(x)$ dan $E(y)$, orang dapat memperoleh $E(x \lozenge y)$ tanpa mendekripsi x dan y untuk beberapa operasi $\lozenge$.

Algoritma enkripsi homomorfik terdiri atas algoritma sifat aditif (penjumlahan) dan multiplikatif (perkalian).

1) Algoritma enkripsi homomorfik aditif:

$$E(x + y) = E(x) \otimes E(y) \quad (3)$$

dengan E adalah fungsi enkripsi, x dan y adalah plainteks, dan $\otimes$ adalah operasi yang bergantung pada *cipher* (misalnya +, ×, atau lainnya).

2) Algoritma enkripsi homomorfik multiplikatif:

$$E(x \bullet y) = E(x) \otimes E(y) \quad (4)$$

dengan E adalah fungsi enkripsi, x dan y adalah plainteks, dan $\otimes$ adalah operasi yang bergantung pada *cipher* (misalnya +, ×, atau lainnya).

Sementara itu, ekripsi homomorfik terdiri atas enkripsi homomorfik sebagian (*partially homomorphic encryption*) dan enkripsi homomorfik penuh (*fully homomorphic encryption*).

1) Enkripsi homomorfik sebagian:

Hanya memungkinkan satu operasi aritmetika pada cipherteks, yaitu operasi penjumlahan (sifat aditif) saja atau perkalian (sifat multiplikatif) saja. Contoh algoritmanya ialah RSA (Rivest-Shamir-Adleman), ElGamal, dan Paillier.

2) Enkripsi homomorfik penuh:

Memungkinkan operasi penjumlahan (sifat aditif) dan perkalian (sifat multiplikatif) sekaligus pada cipherteks. Contoh algoritmanya ialah CKKS (Cheon-Kim-Kim-Song).

C. Algoritma Paillier

Algoritma yang dikembangkan oleh Pascal Paillier ini didasarkan pada sulitnya memecahkan persoalan residu ke-n (*composite residuosity problem*), yaitu sebagai berikut.

"Diberikan bilangan komposit n dan bilangan bulat z. Bilangan z dikatakan residu ke-n modulo n^2 jika terdapat sebuah nilai y sedemikian sehingga $z \equiv y^n \pmod{n^2}$."

Berikut merupakan prosedur pembangkitan pasangan kunci publik dan privat berdasarkan algoritma Paillier.

1) Pilih dua bilangan prima sembarang, p dan q, dengan syarat PBB($p.q$, $(p-1)(q-1) = 1$.
2) Hitung $n = p.q$. (5)
3) Hitung λ = KPK($p-1$, $q-1$). (6)
4) Pilih sembarang bilangan bulat g, dengan syarat $g < n^2$.
5) Hitung $\mu = (L(g^{\lambda} \bmod n^2))^{-1} \bmod n$, (7)
   dengan fungsi $L(x) = (x-1)/n$. (8)

Hasilnya adalah kunci publik (g, n) dan kunci privat (λ, μ). Perhatikan bahwa PBB adalah pembagi bersama terbesar (*greatest common divisor*; GCD) dan KPK adalah kelipatan persekutuan terkecil (*lowest common multiple*; LCM).

Berikut merupakan prosedur enkripsi plainteks p berdasarkan algoritma Paillier, dengan syarat $0 \le p < n$.

1) Pilih bilangan bulat acak r, dengan syarat $0 \le r < n$ dan PBB(r, n) = 1.
2) Hitung cipherteks dengan $c = g^p.r^n \bmod n^2$. (9)
   Dalam hal ini, c adalah residu ke-n dalam modulus n^2, dengan lambang $[c]_g$.

Perhatikan bahwa r dapat berbeda pada setiap enkripsi untuk menjamin bahwa plainteks yang sama tidak akan menghasillkan cipherteks yang sama pula. Hal ini menjamin kerahasiaan data, yaitu bahwa suatu cipherteks tidak dapat dideduksi dari cipherteks lainnya yang serupa.

Berikut merupakan prosedur dekripsi cipherteks c berdasarkan algoritma Paillier: hitung plainteks dengan

$$p = L(c^{\lambda} \bmod n^2).\mu \bmod n \quad (10)$$

atau

$$p = \frac{L(c^{\lambda} \bmod n^2)}{L(g^{\lambda} \bmod n^2)} \bmod n^2 \quad (11)$$

Algoritma Paillier termasuk algoritma enkripsi homomorfik sebagian yang bersifat aditif, artinya hasil dekripsi dari perkalian dua buah cipherteks akan sama dengan hasil penjumlahan kedua plainteks-nya.

$$E(x + y) = E(x).E(y) \quad (12)$$

$$D(E(x).E(y)) = x + y \quad (13)$$

Pembuktian persamaan (12) ialah sebagai berikut.

$$E(m_1, r_1) = c_1 = g^{m1}.r_1^n \bmod n^2$$

$$E(m_2, r_2) = c_2 = g^{m2}.r_2^n \bmod n^2$$
$$E(m_1, r_1).E(m_2, r_2) = c_1.c_2 = g^{m1+m2}.(r_1.r_2)^n \bmod n^2 \quad (14)$$
$$E(m_1 + m_2, r) = g^{m1+m2}.r^n \bmod n^2$$

dengan $r = r_1.r_2$.

Pembuktian persamaan (13) ialah sebagai berikut.

$$D(E(m_1, r_1).E(m_2, r_2)) = D(c_1.c_2)$$
$$= L((c_1.c_2)^\lambda \bmod n^2).\mu \bmod n \quad (15)$$

Substitusikan persamaan (14) ke dalam $(c_1.c_2)^\lambda$.

$$(c_1.c_2)^\lambda = (g^{m1+m2}.(r_1.r_2)^n \bmod n^2)^\lambda$$
$$\equiv g^{(m1+m2)\,\lambda} \bmod n^2 \quad (16)$$
(teorema Carmichael)

$$g^\lambda \equiv (1 + n)^{\lambda[g]_{(1+n)}} \bmod n^2 \quad (17)$$

Substitusikan persamaan (17) ke dalam (16).

$$(c_1.c_2)^\lambda \equiv (1 + n)^{\lambda[g]_{(1+n)}(m1+m2)} \bmod n^2 \quad (18)$$

$$(1 + n)^q \equiv 1 + qn \,(mod\, n^2) \quad (19)$$

Sesuaikan persamaan (18) dengan (19) ke dalam (18).

$$(c_1.c_2)^\lambda \equiv (1 + n)^{\lambda[g]_{(1+n)}(m1+m2)} \bmod n^2$$
$$\equiv 1 + \lambda[g]_{(1+n)}(m_1+m_2)n \,(mod\, n^2) \quad (20)$$

Substitusikan persamaan (20) ke dalam (8) untuk hitung L.

$$L((c_1.c_2)^\lambda \bmod n^2)$$
$$\equiv L(1 + \lambda[g]_{(1+n)}(m_1+m_2)n \,(mod\, n^2))$$
$$\equiv [(1 + \lambda[g]_{(1+n)}(m_1+m_2)n \,(mod\, n^2)) - 1]/n$$
$$\equiv \lambda[g]_{(1+n)}(m_1+m_2) \,(mod\, n) \quad (21)$$

μ adalah balikan dari $\lambda[g]_{(1+n)}$.

$$\mu = (\lambda[g]_{(1+n)})^{-1} \quad (22)$$

Substitusikan (21) dan (22) ke dalam (15).

$$D(c_1.c_2) = L((c_1.c_2)^\lambda \bmod n^2).\mu \bmod n$$
$$\equiv \lambda[g]_{(1+n)}(m_1+m_2) \,(mod\, n).(\lambda[g]_{(1+n)})^{-1} \,(mod\, n)$$
$$= (m_1+m_2) \bmod n \quad (23)$$

III. PEMBAHASAN

Untuk memvisualisasikan pembahasan selanjutnya, akan dibangun platform donasi daring sederhana berbasis web.

A. Kebutuhan Platform

Berdasarkan seluruh fakta di atas, dapat disimpulkan kebutuhan fungsional suatu platform donasi daring berikut.

TABLE I. KEBUTUHAN FUNGSIONAL PLATFORM

No.	Kebutuhan Fungsional
F01	Pengguna dapat melihat daftar kampanye
F02	Pengguna dapat melihat total donasi setiap kampanye
F03	Pengguna dapat melakukan donasi pada kampanye
F04	Pengguna dapat melihat lagi nominal setiap donasinya
F05	Pengguna dapat melihat total donasinya pada platform

B. Perancangan Platform

Platform akan dibangun dengan teknologi web standar saat ini, yaitu Next.js dengan basis data Supabase (relasional berbasis PostgreSQL). Basis data dirancang seperti Fig 2. Terlihat bahwa terdapat tiga tabel, yaitu Users, Campaigns, dan Donations. Users berisi data pengguna, yang tersedia otomatis pada setiap proyek Supabase. Setiap pengguna dapat bertindak sebagai donatur (*donor*), penggalang dana (*fundraiser*), atau keduanya. Pengguna dapat menginisiasi kampanye penggalangan dana (*campaign*) dengan judul, deskripsi, nominal gol, dan *thumbnail* pilihannya. Setiap pengguna juga dapat melakukan donasi (*donation*) pada kampanye pilihannya dengan nominal sesuai keinginannya.

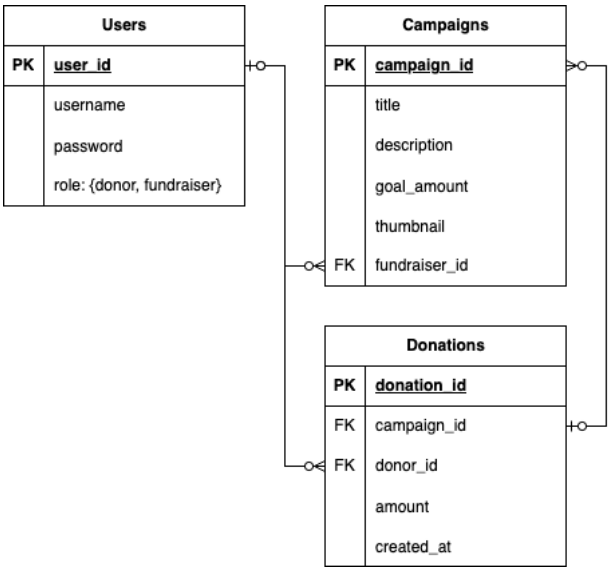

Fig. 2. Desain Basis Data Platform

Supabase

```
create table campaigns (
  campaign_id serial primary key,
  fundraiser id uuid references auth.users not null,
  title varchar(255) not null,
  description text,
  goal_amount int default 0
);

create table donations (
  donation_id serial primary key,
  campaign_id int references campaigns(campaign_id)
    not null,
  donor_id uuid references auth.users not null,
  amount int default 0,
  encrypted_amount text default "0",
  created_at timestamp default now()
);

alter table donations enable row level security;

create policy "Individuals can view donations" on
donations for
  select using (auth.uid() is not null);
```

```
create policy "Individuals can create donations" on
donations for
  insert with check (auth.uid() = donor_id);
```

Skema pertukaran data dirancang seperti Fig 3. Terlihat bahwa algoritma Paillier hanya diimplementasikan pada *server*, sehingga operasi penjumlahan, enkripsi, dan dekripsi hanya dieksekusi oleh *server*. Ketika pengguna ingin melihat data, *server* akan mendekripsinya terlebih dahulu dengan kunci privatnya (F04), atau menjumlahkannya terlebih dahulu baru mendekripsinya (F02 dan F05), sedangkan ketika pengguna ingin menambah data, *server* akan mengenkripsinya terlebih dahulu dengan kunci publiknya (F03).

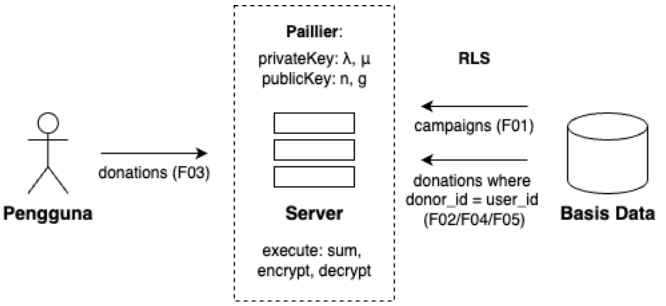

Fig. 3. Skema Pertukaran Data

C. Implementasi Algoritma Paillier

Untuk mempermudah, algoritma Paillier diimplementasi dengan sejumlah fungsi dari pustaka (*library*) bigint-crypto-utils, yaitu gcd untuk menghitung FPB, lcm untuk menghitung KPK, modInv untuk menghitung invers modulo, modPow untuk menghitung modulo dengan basis bereksponen, randBetween untuk menghasilkan bilangan acak antara dua buah bilangan, dan prime untuk menghasilkan sebuah bilangan prima acak. Hasilnya ialah sejumlah fungsi baru, yaitu generatePQ untuk menghasilkan P dan Q acak sesuai syarat, generatePublicKey untuk menghasilkan kunci publik n dan g, generatePrivateKey untuk menghasilkan kunci privat λ dan μ, generateRandomR untuk menghasilkan R acak sesuai syarat, encrypt untuk mengenkripsi plainteks, decrypt untuk mendekripsi plainteks, serta addition untuk menjumlahkan dua cipherteks. Selengkapnya silakan buka repositori GitHub, tepatnya dalam *folder* src/utils/algorithm/paillier.ts.

Sebagai catatan, sebenarnya algoritma Paillier tidak perlu diimplementikan mandiri, karena telah tersedia pustaka yang efektif oleh juanelas [4]. Pustaka ini telah menyediakan fungsi generateRandomKeys untuk membangkitkan pasangan kunci publik dan privat, encrypt untuk mengenkripsi plainteks, decrypt untuk mendekripsi cipherteks, dan addition untuk menjumlahkan dua buah plainteks dalam bentuk cipherteks.

IV. Hasil dan Pembahasan

Secara umum, platform telah berhasil memenuhi seluruh kebutuhan fungsional yang telah didefinisikan sebelumnya. Selengkapnya silakan buka tautan web.

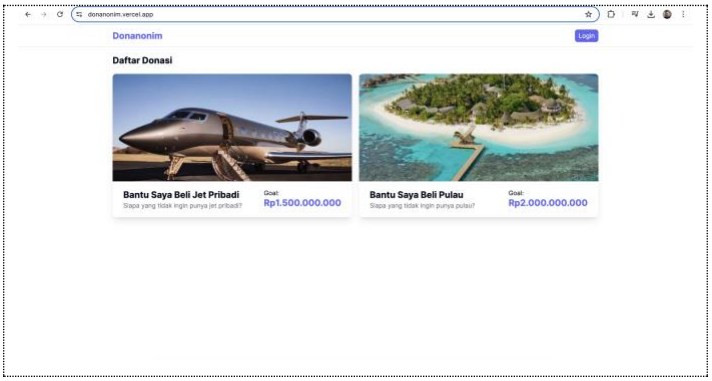

Fig. 4. Melihat Daftar Kampanye (F01)

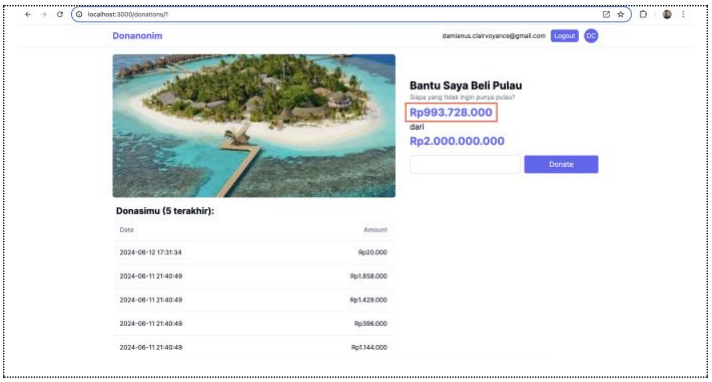

Fig. 5. Melihat Total Donasi Setiap Kampanye (F02)

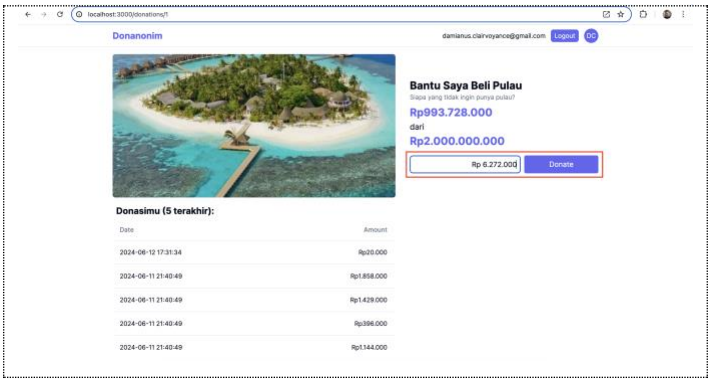

Fig. 6. Melakukan Donasi pada Kampanye (F03)

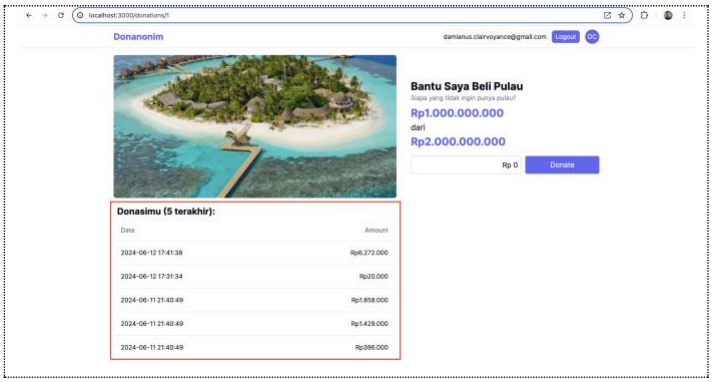

Fig. 7. Melihat Lagi Nominal Setiap Donasi (F04)

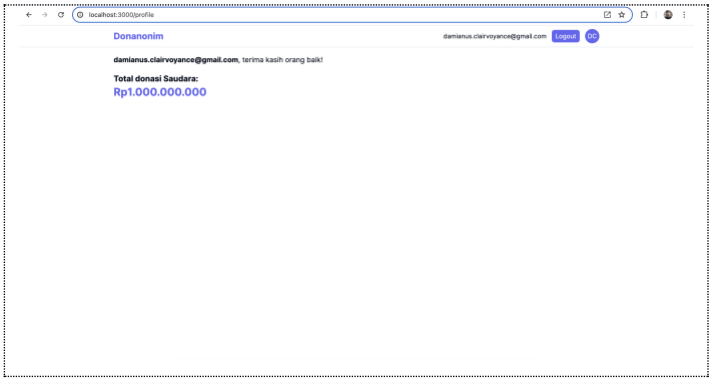

Fig. 8. Melihat Total Donasi pada Platform (F05)

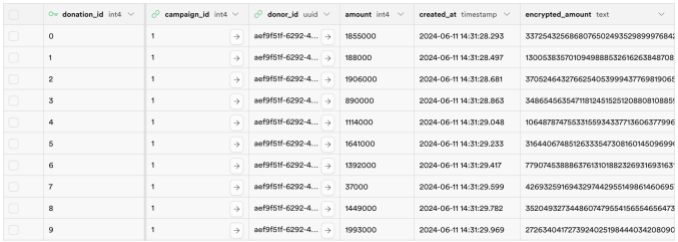

Fig. 9. Nominal Terenkripsi dalam Tabel Donasi

Perhatikan bahwa pada Fig 9, tersedia data nominal donasi dalam cipherteks (encrypted_amount) dan plainteks (amount). Data plainteks hanya merupakan bentuk pengujian untuk makalah ini, sementara dalam kasus nyata, kolom amount dapat dihapus sepenuhnya.

A. Keamanan Algoritma

Untuk menciptakan keamanan setingkat 128-bit, panjang kunci yang dibangkitkan harus setidaknya 3072 bit. Namun karena pada dasarnya data yang dirahasiakan tidak krusial, panjang kunci dapat divariasikan, misalnya 2048 atau 1024 bit. Perlu diingat bahwa untuk memecahkan keamanan 128-bit dengan *brute force*, penyerang memerlukan 2^{128} percobaan. Sebagai bayangan, jika suatu sistem seribu komputer dapat menebak masing-masing 1 miliar (10^9) kunci per detik, sistem ini memerlukan waktu 10^{19} tahun.

B. Kecepatan Algoritma

Sebagai bentuk penilaian kinerja, kecepatan algoritma Paillier akan dibandingkan terhadap sejumlah tolok ukur. Perbandingan pertama ialah membandingkan kecepatan penjumlahan cipherteks dengan algoritma Paillier dan penjumlahan plainteks biasa dalam tiga ukuran data berbeda, yaitu 10, 100, dan 1.000, serta dalam panjang kunci 2048 bit.

TABLE II. PENJUMLAHAN CIPHERTEKS (PAILLIER) VS. PLAINTEKS

Data	Cipherteks (Paillier) [dalam s]	Plainteks [dalam s]
10	0.198	0.000071
100	0.199	0.000099
1.000	0.247	0.000155

Perhatikan bahwa penjumlahan cipherteks berjalan lebih lama daripada penjumlahan plainteks dalam orde ribuan kali lipat. Namun, waktu eksekusi penjumlahan cipherteks tidak berkembang secara linear ataupun eksponensial sehingga memungkinkan untuk menjumlahkan banyak data berapa pun dengan waktu eksekusi yang kurang lebih sama.

Perbandingan kedua ialah membandingkan kecepatan penjumlahan cipherteks algoritma Paillier dalam tiga panjang kunci berbeda, yaitu 1024, 2048, dan 3072 bit. Perhitungan didasarkan pada waktu untuk penjumlahan 100 data. Terbukti bahwa penggunaan kunci 1024 alih-alih 2048 bit meningkatkan efisiensi algoritma, meskipun mengorbankan keamanan.

TABLE III. PENJUMLAHAN DENGAN KUNCI 1024 VS. 2048 VS. 3072 BIT

	1024 bit [dalam s]	2048 bit [dalam s]	3072 bit [dalam s]
Kecepatan	0.039	0.211	0.579

Perbandingan terakhir ialah membandingkan kecepatan enkripsi plainteks dan dekripsi cipherteks dalam tiga panjang kunci berbeda, yaitu 1024, 2048, dan 3072 bit. Perhitungan didasarkan pada rerata waktu untuk 100 operasi. Terlihat bahwa *server* hanya memerlukan waktu sekitar 0.1 detik untuk melakukan enkripsi dan dekripsi data dengan kunci 2048 bit.

TABLE IV. ENKRIPSI DAN DEKRIPSI DENGAN KUNCI 1024 VS. 2048 VS. 3072 BIT

	1024 bit [dalam s]	2048 bit [dalam s]	3072 bit [dalam s]
Enkripsi	0.033	0.185	0.544
Dekripsi	0.032	0.182	0.534

KESIMPULAN

Enkripsi homomorfik dapat diimplementasikan dalam platform donasi daring, dengan kebutuhan fungsional utama berupa penjumlahan nominal donasi dalam suatu penggalangan dana. Perlu dicatat pula penjumlahan cipherteks hanya memakan waktu kurang lebih 0.2 detik (dalam sistem penyusun) dan tidak berkembang secara linear ataupun eksponensial. Oleh karena itu, sistem ini dapat dimanfaatkan untuk berbagai penggalangan dana, tetapi untuk skala yang lebih besar, tetap disarankan untuk menyimpan nilai total donasi sebagai atribut sendiri pada tabel basis data `campaigns` dan `users`, sehingga *server* tidak perlu menghitung penjumlahan cipherteks setiap saat.

PENUTUP

Penyusun mengucapkan puji syukur kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penyusun dapat menyelesaikan makalah ini. Penyusun juga tak lupa berterima kasih kepada semua pihak yang terlibat dalam penyusunan makalah ini, baik secara langsung maupun tidak. Penyusun mengucapkan terima kasih kepada Bapak Rinaldi Munir selaku dosen pengampu mata kuliah IF4020 Kriptografi Semester II Tahun 2023/2024 yang telah membimbing, serta kepada seluruh keluarga dan teman penulis yang telah mendukung.

Penyusun menyadari bahwa masih terdapat banyak kekurangan dalam penyusunan makalah ini. Oleh karena itu, penulis menerima secara terbuka setiap umpan balik yang membangun dengan membuka ruang diskusi terkait topik ini melalui pos-el. Akhir kata, penyusun berharap makalah ini dapat mendatangkan manfaat bagi banyak orang.

REPOSITORI GITHUB

Kode platform web dan algoritma Paillier, beserta penilaian kinerjanya (terpisah dalam *folder* `evaluation`), dapat diakses melalui tautan berikut.

https://github.com/dclairvoyance/Makalah_IF4020_Kripto.git

PLATFORM WEB

Platform web dapat diakses melalui tautan berikut.

https://donanonim.vercel.app/

REFERENSI

[1] Muhamad, N. (2023, November 16). *10 Negara Paling Dermawan di Dunia 2023, Indonesia Juara*. Katadata. https://databoks.katadata.co.id/datapublish/2023/11/16/10-negara-paling-dermawan-di-dunia-2023-indonesia-juara

[2] Annur, C. M. (2023, March 14). *Banyak Orang Indonesia Sering Donasi Online Lebih dari 2,5% Penghasilannya*. Katadata. https://databoks.katadata.co.id/datapublish/2023/03/14/banyak-orang-indonesia-sering-donasi-online-lebih-dari-25-penghasilannya

[3] National Council of Nonprofits. (n.d.). *Principles & Practices: "Best Practices" for Nonprofits*. National Council of Nonprofits. https://www.councilofnonprofits.org/running-nonprofit/governance-leadership/principles-practices-best-practices-nonprofits

[4] Juanelas. (n.d.). GitHub - juanelas/paillier-bigint: An implementation of the Paillier cryptosystem using native JS implementation of BigInt. GitHub. https://github.com/juanelas/paillier-bigint

[5] Munir, R. (2024). *Kriptografi Kunci-Publik – Bahan Kuliah IF4020 Kriptografi [salindia]*. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2023-2024/17-Kriptografi-Kunci-Publik-2024.pdf

[6] Munir, R. (2024). *Enkripsi Homomorfik – Bahan Kuliah IF4020 Kriptografi [salindia]*. https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2023-2024/38-Enkripsi-homomorfik-2024.pdf

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, 12 Juni 2024

Damianus Clairvoyance Diva Putra