Penerapan *Blockchain* dan Kriptografi untuk Keamanan Data pada Jaringan *Smart Grid*

Hafizh Fianto Putra, Wirawan, dan Ontoseno Penangsang
Departemen Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: ontosenop@ee.its.ac.id

***Abstrak*—*Smart city* adalah sebuah konsep kota yang mengandalkan teknologi *Internet of Things* (IoT) untuk membantu meningkatkan kualitas kota tersebut. *Smart grid* adalah salah satu komponen dari *smart city* yang memperkenalkan komunikasi dua arah antara pelanggan dengan perusahaan penyedia listrik. Salah satu masalah yang dapat terjadi pada jaringan *smart grid* adalah data pelanggan yang jatuh ke pihak yang tidak bertanggungjawab karena saluran transmisi yang tidak aman. Oleh karena itu, penggunaan *blockchain* dan kriptografi dapat diterapkan pada jaringan *smart grid* untuk menyelesaikan masalah tersebut. *Blockchain* memberikan keamanan tambahan untuk penyimpanan data pada pusat pengatur jaringan dan kriptografi memberikan kerahasiaan dan autentikasi pada pertukaran data dalam jaringan. Sistem yang baru diciptakan ini harus bisa menyesuaikan frekuensi pengambilan data pada jaringan *smart grid*. Penelitian ini bertujuan untuk mengetahui waktu komputasi yang dibutuhkan dari kedua proses tersebut dengan menyusun suatu simulasi jaringan *smart grid*. Simulasi dalam penelitian ini dilakukan dengan menggunakan aplikasi MATLAB dan tersusun dari dua program utama, yaitu program kriptografi menggunakan algoritma RSA dan program pembuatan *blockchain* sederhana. Hasil yang didapatkan menunjukkan bahwa rentang waktu komputasi RSA pada suatu jaringan *smart grid* lebih cepat dari frekuensi pengambilan data yang ditentukan, yaitu sebesar satu menit. Sementara itu, rentang waktu penyusunan *blockchain* dengan ketentuan tertentu tidak dapat memenuhi persyaratan yang sama.**

Kata Kunci*—*Blockchain*, Kriptografi, MATLAB, RSA, *Smart Grid.

I. PENDAHULUAN

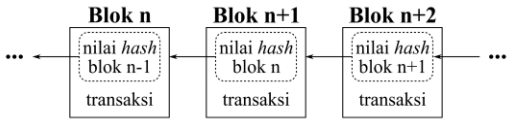
SMART grid adalah sebuah jaringan nasional yang menggunakan teknologi informasi pada kelistrikan agar efisien, dapat diandalkan, dan aman [1]. *Smart grid* sejatinya adalah salah satu komponen dari *smart city*, kota yang menerapkan *Internet of Things* (IoT) untuk meningkatkan kualitas kota dan rakyat yang tinggal di kota tersebut. Pada jaringan *smart grid*, akan ada pertukaran data secara dua arah antara pengguna listrik dengan perusahaan penyedia listrik. Contoh data yang dapat diambil dari meteran listrik pengguna adalah nilai tegangan, arus, dan *power factor* [2].

Sebagai contoh, Umass Smart yang dikeluarkan oleh salah satu laboratorium di University of Massachusetts Amherst mengoleksi data tiap satu menit berupa sepuluh digit *unix timestamp* dan nilai *watt* yang terukur oleh meteran listrik [3]. Sementara itu, Customer Behavior Trials (CBT) yang dikeluarkan oleh badan regulator di Irlandia, Commission for Regulation of Utilities (CRU) menunjukkan tiga informasi pada hasil pengukuran *smart meter* yang berupa nomor identifikasi meteran, kode yang menunjukkan tanggal dan waktu, dan konsumsi listrik selama interval 30 menit [4].

Masalah yang dapat terjadi pada *smart grid* adalah data pelanggan yang jatuh ke pihak yang tidak bertanggungjawab karena saluran transmisi yang tidak aman [5]. Data yang didapatkan tersebut dapat disalahgunakan sehingga merugikan pelanggan maupun perusahaan penyedia listrik.

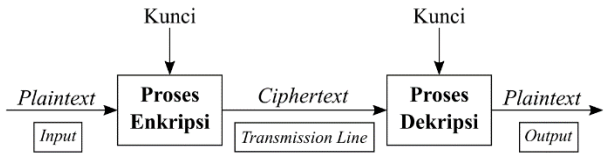
Penggunaan *blockchain* dan kriptografi dapat diterapkan pada jaringan *smart grid* untuk menyelesaikan masalah tersebut. *Blockchain* merupakan sebuah rekaman autentik dari seluruh aktivitas yang terjadi dalam suatu jaringan, tersebar di seluruh komponen jaringan, dan saling terkait dalam suatu rangkaian blok data [6]. *Blockchain* dapat diasumsikan sebagai arsip transaksi yang dikumpulkan pada blok-blok dengan penanda waktu atau *timestamp*. Setiap blok juga teridentifikasi dengan suatu nilai *hash*. Namun, setiap blok tersebut mereferensikan nilai *hash* dari blok yang ada sebelumnya.

Fungsi *hash* merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap. Fungsi ini digunakan pada berbagai aplikasi pengamanan dan protokol internet. Beberapa contoh kegunaannya adalah untuk autentikasi pesan, tanda tangan digital, dan penyimpanan *password* [7].



Gambar 1. Model sederhana *blockchain*.

Sistem kriptografi diterapkan untuk mengamankan data yang dikirimkan sehingga data tersebut hanya bisa diakses oleh pengirim dan penerima. Kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan tersebut dikirim dari suatu tempat ke tempat lain [8].



Gambar 2. Model sederhana kriptografi.

Skema yang digunakan pada penelitian ini adalah kriptografi asimetris bernama RSA. RSA memodelkan *plaintext* dan *ciphertext* menjadi sebuah bilangan bulat [9]. RSA memanfaatkan prinsip dari eksponensial dan aritmetika modular. *Plaintext* dienkripsi dalam blok-blok yang

mempunyai ukuran dengan nilai biner tertentu. Setelah itu, blok data diolah menggunakan rumus berikut:

$$C = M^e \bmod (n) \;;\; M = C^d \bmod (n) \quad (1)$$

C adalah blok data *ciphertext* dan *M* adalah blok data *plaintext*. Pengirim dan penerima mengetahui nilai dari *n* dan *e*. Namun, nilai *d* hanya diketahui oleh penerima. Oleh karena itu, RSA ini berupa algoritma enkripsi asimetris atau kunci-publik dengan kunci umum *PU* = [*e, n*] dan kunci privat *PR* = [*d, n*].

Nilai *n* didapatkan dari perkalian bilangan prima p dan q dengan $p \neq q$. Lalu, $\Phi(n)$ dihitung dengan rumus $\phi(n) = (p-1)(q-1)$. Nilai e dipilih sehingga memenuhi $\text{FPB}(\phi(n), e) = 1$ dan $1 < e < \phi(n)$. Sedangkan nilai *d* dicari sehingga memenuhi $e \cdot d \equiv 1 \left(\text{mod } (\phi(n))\right)$.

Salah satu variabel penting yang terdapat pada RSA adalah pemilihan ukuran kunci yang digunakan. Ukuran kunci yang terlalu kecil akan memudahkan sistem untuk bisa dibobol oleh pihak yang tidak bertanggung jawab. Oleh karena itu, National Institute of Standards and Technology (NIST) memberikan rekomendasi untuk tidak menggunakan kunci yang berukuran 1024 bit atau yang lebih kecil untuk pengamanan informasi penting di pemerintahan [10].

Sistem yang baru diciptakan tersebut harus bisa menyesuaikan frekuensi pengambilan data pada jaringan *smart grid*. Penelitian ini akan merancang suatu simulasi dari jaringan *smart grid* secara sederhana dengan menerapkan *blockchain* dan kriptografi untuk mengetahui pengaruh dari variabel yang terdapat pada keduanya. Simulasi dilakukan dengan menggunakan aplikasi MATLAB, yang tersusun dari program kriptografi menggunakan algoritma RSA dan program pembuatan *blockchain* sederhana.

Penelitian ini memiliki beberapa batasan atau asumsi untuk mempermudah penyelesaian masalah, seperti saluran transmisi dianggap bersifat sempurna atau tidak menimbulkan galat pada data yang ditransmisikan, penggunaan skema RSA, jaringan mempunyai jumlah klien tetap, dan simulasi tidak dilakukan menggunakan perangkat *smart meter* yang sebenarnya, tetapi hanya menggunakan laptop dan komputer.

Tujuan utama dari penelitian ini adalah mendapatkan waktu komputasi yang dibutuhkan oleh program *blockchain* dan kriptografi pada suatu model jaringan *smart grid*. Selain itu, penelitian ini bertujuan untuk mengetahui pengaruh lainnya yang dapat ditimbulkan oleh perubahan variabel yang terdapat pada *blockchain* dan algoritma RSA.

Penelitian ini diharapkan dapat memberikan manfaat seperti pemahaman kriptografi dan aplikasinya pada dunia nyata. Selain itu, penelitian ini ingin membuktikan bahwa *blockchain* bukanlah topik yang terlalu susah dan kompleks untuk dipelajari oleh mahasiswa tingkat sarjana. Di sisi lain, *smart grid* juga merupakan contoh baik untuk penelitian karena dapat menggabungkan dua jenis bidang studi, yaitu bidang telekomunikasi dan bidang sistem tenaga.

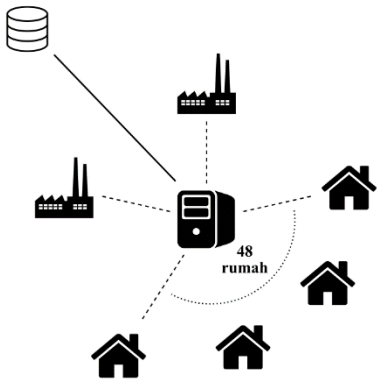
II. URAIAN PENELITIAN

A. Peninjauan Pustaka

Peninjauan pustaka dilakukan untuk mencari kondisi terkini dari topik yang sejenis dengan penelitian ini. Beberapa topik seperti kriptografi, *blockchain*, dan *smart grid* telah menghasilkan banyak *paper* yang telah dipublikasikan. Sehingga, penelitian ini dapat menjadi tambahan atau pelengkap dari penelitian yang telah dilakukan.

B. Desain Sistem

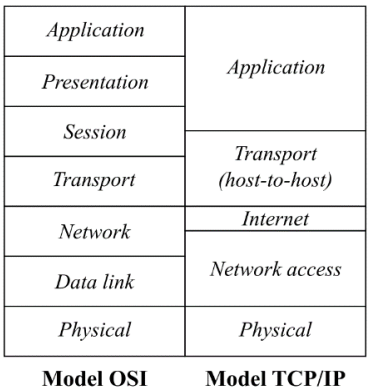
Program awalnya disusun berdasarkan dua fungsi utama yang akan diterapkan pada jaringan, yaitu fungsi RSA dan fungsi *hash*. Kedua fungsi ini berguna untuk mengamati pengaruh variabel penting pada fungsi terhadap waktu komputasi yang dibutuhkan untuk satu pengguna. Waktu yang diamati adalah lamanya fungsi untuk mengolah satu nilai masukan hingga menghasilkan keluaran.



Gambar 3. Desain jaringan *smart grid* lokal yang disimulasikan.

Setelah itu, sebuah jaringan *smart grid* dirancang sebagai asumsi penerapan program yang disusun. Jaringan tersusun dari 48 rumah dan dua pabrik yang terhubung dengan *server* lokal. Selain itu, *server* lokal terhubung dengan sebuah *database*. *Database* ini berperan dalam penyimpanan data yang menggunakan sistem *blockchain*. Topologi jaringan yang digunakan pada penelitian ini adalah ***star topology*** atau topologi bintang. Topologi jenis ini dipilih karena penelitian ini diasumsikan hanya pada skala lokal.

Pada komunikasi antara dua perangkat, terdapat sebuah protokol yang harus dipenuhi. Hal itu dikarenakan prosedur yang terlibat dapat menjadi rumit sehingga kesepakatan tingkat tinggi dibuat di antara dua perangkat. Salah satu arsitektur yang sering digunakan adalah Transmission Control Protocol dan Internet Protocol yang membentuk rangkaian protokol TCP/IP. Protokol ini yang menjadi dasar dari sistem internet. Arsitektur lainnya yang cukup dikenal tetapi jarang digunakan adalah model Open Systems Interconnection (OSI) [11].

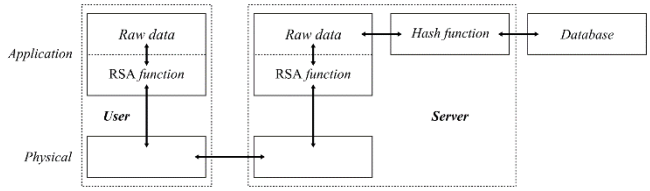


Gambar 4. Perbandingan *layer* pada model OSI dan model TCP/IP.

Program yang dibuat dalam penelitian ini akan menjadi salah satu komponen dalam jaringan *smart grid*. Karena jaringan tersebut bekerja layaknya jaringan telekomunikasi lainnya, program akan dirancang untuk bisa kompatibel dengan desain jaringan telekomunikasi yang sudah ada. Program pertama yang berupa fungsi RSA diletakkan dekat dengan ***raw data*** atau data kasar. Lalu, program kedua yang berupa fungsi *hash* diletakkan sebelum proses penyimpanan oleh *server*.

C. Perancangan Program

Program yang telah berjalan dengan baik nantinya akan diuji dengan menjalankan program secara berulang-ulang. Hal tersebut dimaksudkan untuk mendapatkan rangkuman hasil dari setiap jenis program yang telah dibuat.



Gambar 5. Skema lengkap dari penerapan program pada komunikasi dengan prinsip *end-to-end* antara *user* dan *server*.

Pada program RSA yang dibuat untuk jaringan *smart grid* ini, program terdiri dari beberapa informasi dan variabel yang didefinisikan terlebih dahulu, yaitu:

1. *Raw data string* yang memuat beberapa informasi yang dipisahkan oleh tanda pagar. Informasi tersebut memiliki panjang maksimal 32 karakter yang terdiri dari kode identifikasi pelanggan, *unix time*, pemakaian listrik, dan daya terukur. Contoh *string* yang dihasilkan:
   `AP24V0#1543509173#123456#1234.56`
2. Pengambilan data pada jaringan *smart grid* diasumsikan akan dilakukan setiap satu menit.
3. Ukuran kunci RSA yang digunakan untuk rumah adalah 1536 bit dan untuk pabrik adalah 3072 bit.
4. Keluaran dari fungsi enkripsi berupa matriks baris dari bilangan yang sebanding dengan kunci. Sehingga, untuk rumah berukuran 1 x 192 x 8-bit *signed integer* dan pabrik berukuran 1 x 384 x 8-bit *signed integer*.
5. Satu siklus didefinisikan dengan pengiriman data dari tiap pengguna menuju server dan sebaliknya. Untuk pengiriman dari *server*, *data string* memuat kode identifikasi pelanggan, kode perintah, dan karakter pengisi dengan jumlah karakter maksimal 32 karakter. Contoh *string* yang dihasilkan:
   `AP24X1#25#XXXXXXXXXXXXXXXXXXXXXXXX`

Sedangkan program *hash* yang dibuat untuk jaringan *smart grid* merepresentasikan proses penyusunan blok dari *blockchain* oleh *server*. Program ini disendirikan karena program ini akan memerlukan waktu yang lebih lama jika dibandingkan dengan program sebelumnya. Hal ini dikarenakan adanya ketentuan yang dibuat untuk nilai *hash* yang diinginkan. Sebagai contoh, nilai *hash* berikut yang memenuhi ketentuan tiga digit nol:

`000B5B89851DA6EDB6427C25D5CCA7600F189ADF002C737B1A29E1E060ADF56F`

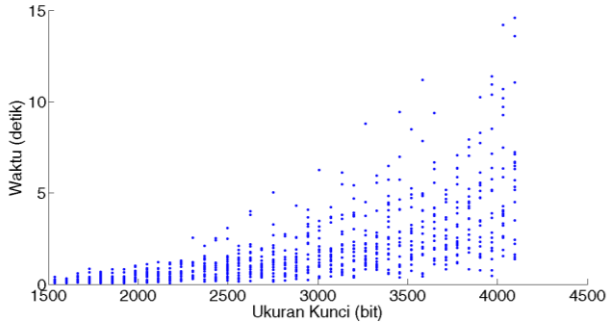
Kegunaan nilai *hash* yang harus memenuhi suatu ketentuan bertujuan sebagai *proof of work* sebagai bukti bahwa pekerjaan untuk menghitung nilai *hash* telah dilakukan untuk sebuah masukan *string*.

Proses yang dilakukan pada kode program fungsi *hash* ini hampir sama dengan cara pada transakasi Bitcoin, tetapi sistem pada Bitcoin mengharuskan nilai *hash* yang dihasilkan berada di bawah suatu nilai atau dikenal dengan istilah ***target***. Sebagai perbandingan, nilai *target* saat ini memiliki awalan nol untuk 18 digit pertama [12].

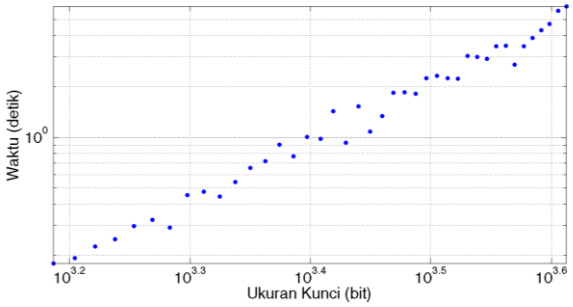
III. ANALISIS DAN PEMBAHASAN DATA

A. Ukuran Kunci RSA

Gambar 6 memberikan gambaran persebaran data yang didapatkan untuk beberapa ukuran kunci RSA. Setiap nilai ukuran kunci dilakukan pengukuran sebanyak 20 kali. Sedangkan rentang nilai ukuran kunci yang diuji adalah dari 1536 bit hingga 4092 bit dengan interval 64 bit. Dari Gambar 6, terlihat bahwa dengan semakin besarnya ukuran kunci mengakibatkan waktu komputasi cenderung lebih bervariasi. Selain itu, pergerakan nilai rata-rata waktu komputasi menunjukkan tren naik di setiap ukuran kunci. Meskipun terjadi fluktuasi pada kenaikan yang terjadi, hal tersebut masih dapat menjadi bukti bahwa ukuran kunci RSA berpengaruh secara eksponensial terhadap waktu komputasi yang dibutuhkan.



Gambar 6. Grafik persebaran waktu komputasi beberapa ukuran kunci RSA.



Gambar 7. Grafik "log-log" rata-rata waktu komputasi setiap ukuran kunci.

B. Ketentuan Nilai Hash

Pengujian ketentuan nilai *hash* dilakukan dengan menghitung waktu yang dibutuhkan untuk mencari satu nilai *hash* yang memenuhi ketentuan berupa jumlah digit nol yang mengawali nilai *hash* tersebut. Jumlah digit nol yang dipilih sebanyak dua, tiga, dan empat digit. Setiap ketentuan dijalankan sebanyak 250 kali.

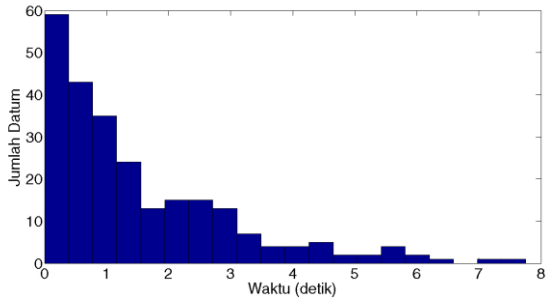
Tabel 1.
Rangkuman pengaruh ketentuan nilai *hash* terhadap waktu komputasi

Nama Variabel	2 Digit Nol	3 Digit Nol	4 Digit Nol
Waktu Minimum	0,001114	0,000808	0,470296
Rata-rata	0,096102	1,53117	21,98469
Waktu Maksimum	0,458466	7,751967	137,1826
Simpangan Baku	0,090674	1,517616	20,37014

*angka dalam detik

Dari data yang didapatkan pada Tabel 1, terlihat bahwa dengan semakin banyaknya digit nol yang menjadi ketentuan awalan nilai *hash*, rata-rata waktu komputasi yang dibutuhkan juga semakin lama. Selain itu, dengan bertambahnya satu digit nol, rata-rata waktu yang didapatkan juga terlihat naik dengan kelipatan tertentu.

Gambar 8 menunjukkan model persebaran data yang didapatkan untuk fungsi *hash* ini bersifat condong kanan (*skewed right*). Sifat tersebut menunjukkan data memiliki "ekor" kanan yang lebih panjang. Dengan kata lain, distribusi data pada sifat ini terpusat di sebelah kiri grafik dan lebih banyak datum yang berada di bawah nilai rata-rata data.



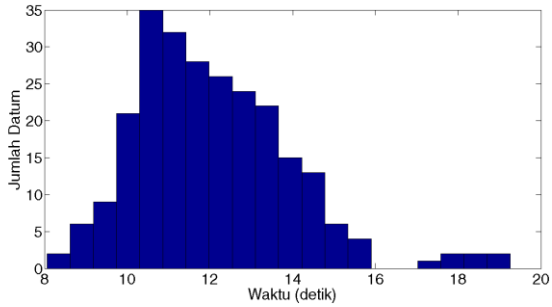
Gambar 8. Histogram untuk ketentuan tiga digit nol.

C. Kinerja Program RSA

Program RSA dalam penelitian ini tersusun oleh 50 kali proses enkripsi dan dekripsi untuk pengiriman informasi listrik pelanggan dari pengguna ke *server* beserta 50 kali proses yang sama untuk pengiriman kembali informasi perintah dari *server* ke pengguna. Jumlah proses tersebut menyesuaikan jumlah pengguna yang berada dalam jaringan yang diasumsikan.

Dalam penelitian yang telah dilakukan, program berhasil melakukan proses enkripsi dan dekripsi dari seluruh data. Hal tersebut dikarenakan pesan yang dikirimkan sudah sama persis dengan pesan yang diterima.

Program RSA telah dijalankan sebanyak 250 kali dan hasilnya terangkum pada Gambar 9 dan Tabel 2. Gambar 9 menunjukkan bahwa persebaran data yang didapatkan selama simulasi bersifat condong kanan dengan kelompok datum terbanyak berada di kisaran 10,5 detik.



Gambar 9. Histogram simulasi program RSA.

Tabel 2.
Rangkuman data simulasi program RSA

Waktu Minimum	Rata-rata	Waktu Maksimum	Simpangan Baku
8,0604 detik	12,0723 detik	19,2570 detik	1,9086 detik

Jika pengambilan data pada jaringan *smart grid* diasumsikan akan dilakukan setiap satu menit, program RSA ini dapat dinyatakan layak untuk diterapkan pada model jaringan yang telah ditentukan. Hal itu dikarenakan rentang waktu yang dihasilkan oleh program masih lebih cepat dari asumsi frekuensi pengambilan data yang ditentukan.

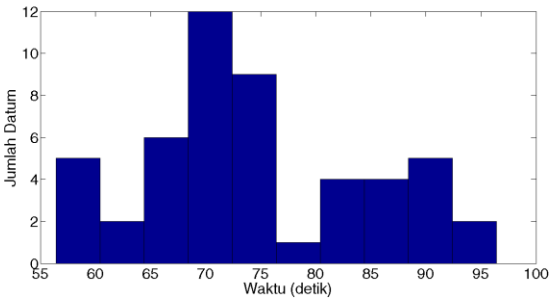
D. Kinerja Program Blockchain

Program *blockchain* dalam penelitian ini tersusun oleh 50 buah *string* yang disusun untuk menghasilkan *blockchain* sederhana. Nilai *hash* dari blok pertama atau *genesis block* diatur menjadi seluruhnya berupa nol sebanyak 64 digit. Nilai tersebut menjadi acuan untuk pembentukan blok selanjutnya.

Dalam penelitian yang telah dilakukan, program berhasil untuk menyusun *blockchain* sederhana. Sebagai contoh, berikut ini adalah tiga blok pertama yang menjadi keluaran program:

```
0000000000000000000000000000000000#UWTYK6#1
541955600#437.57#2335.16#GLLL44B3E2A0PO7F
OAR8JDXRU37A4I
00044A4ACE6E68DFDBE76021CB51B921#YJDK4X#1
541955600#60.782#331.027#OQURQNWPW1LC8OFH
JPOXAR5C5F55EI
000C3530C0BF3D217D24A8D509D84CF2#MES6WX#1
541955600#4.9992#57.3303#NO2D9T211KPSELGY
JC29V1IBCO0D7U
```

Setiap blok tersebut tersusun dari: 32 digit pertama dari nilai *hash* blok sebelumnya; *raw data string* yang tersusun atas kode pelanggan, kode waktu, besar pemakaian listrik, dan besar daya listrik; terakhir adalah tambahan karakter acak untuk pemenuhan ketentuan nilai *hash* (nol pada tiga digit pertama). Secara keseluruhan, satu blok dalam *blockchain* sederhana ini tersusun dari 96 karakter termasuk tanda pagar.



Gambar 10. Histogram simulasi program *blockchain.*

Gambar 10 memberikan gambaran persebaran data yang didapatkan selama simulasi program *blockchain* pada jaringan. Program hanya dijalankan sebanyak 50 kali dikarenakan kepastian yang didapatkan dari hasil simulasi program *hash* untuk tiap individu sebelumnya. Program *blockchain* yang digunakan pada jaringan menggunakan ketentuan tiga digit nol pada awal nilai *hash*. Waktu rata-rata yang didapatkan untuk tiap individu adalah sekitar 1,53 detik. Jika waktu tersebut dikalikan dengan 50, hasilnya adalah sekitar 75 detik mendekati rata-rata waktu komputasi yang terdapat pada Tabel 3. Waktu

tersebut telah melampaui asumsi pengambilan data pada jaringan yang dilakukan setiap satu menit.

Tabel 3.
Rangkuman data simulasi program *blockchain*

Waktu Minimum	Rata-rata	Waktu Maksimum	Simpangan Baku
56,4208 detik	74,6123 detik	96,3993 detik	10,4414 detik

Namun, lamanya waktu menjalankan program *blockchain* tersebut tidak bisa menggambarkan keadaan yang sebenarnya jika diterapkan secara langsung pada suatu jaringan *smart grid*. Hal tersebut dikarenakan program penyusunan *blockchain* dapat dilakukan oleh perangkat yang mempunyai spesifikasi lebih baik dari komputer pada umumnya. Perangkat tersebut memang dikhusukan untuk melakukan pekerjaan dengan intensitas tinggi seperti pada *server*.

IV. KESIMPULAN/RINGKASAN

A. *Kesimpulan*

Ukuran kunci pada sistem kriptografi RSA berpengaruh secara eksponensial terhadap waktu komputasi yang dibutuhkan. Sehingga, besarnya ukuran kunci yang terlalu besar tidak disarankan untuk digunakan secara praktis. Namun, ukuran kunci yang terlalu kecil juga tidak disarankan karena kemungkinan pembobolan menjadi lebih tinggi.

Perhitungan nilai *hash* pada pembentukan *blockchain* membutuhkan waktu yang semakin lama seiring dengan semakin sulitnya ketentuan yang perlu dipenuhi. Pengaruh penambahan satu digit yang berulang pada ketentuan mengakibatkan perubahan yang signifikan terhadap waktu pencarian nilai *hash*.

Penerapan program RSA dan *blockchain* dapat diterapkan pada model jaringan *smart grid* yang memiliki topologi bintang atau pohon. Di bidang kelistrikan, distribusi listrik pada *smart grid* menggunakan pola radial.

Peletakan program RSA pada *application layer* memungkinkan program dapat diletakkan pada berbagai jenis jaringan seperti TCP/IP. Selain itu, pembentukan *blockchain* yang hanya dikhusukan pada suatu perangkat memungkinkan adanya pembagian tanggung jawab dalam penggunaan data.

Kriptografi menggunakan RSA dapat memberikan keamanan pada jaringan *smart grid* dengan merahasaikan transmisi data antara pelanggan dan *server*. Di sisi lain, pembentukan *blockchain* dapat memberikan keamanan tambahan saat penyimpanan data dalam *database*.

Simulasi program yang telah dijalankan perlu dikembangkan dengan lebih lanjut pada jaringan *smart grid* lainnya karena adanya perbedaan perangkat dan aplikasi yang akan digunakan. Aplikasi MATLAB yang dipakai pada penelitian ini tidak bisa diterapkan secara langsung pada beberapa perangkat. Selain itu, jenis perangkat yang digunakan pada jaringan *smart grid* akan memiliki spesifikasi yang berbeda dengan perangkat yang digunakan selama penelitian.

B. *Saran*

Pengujian sistem kriptografi lainnya seperti Advanced Encryption Standard (AES) dapat dilakukan untuk mengetahui waktu komputasi yang dibutuhkan.

Jenis perangkat yang digunakan perlu lebih menyesuaikan dengan kondisi sebenarnya pada suatu jaringan *smart grid*.

Program perlu disusun ulang dengan tidak menggunakan MATLAB agar bisa diterapkan di lebih banyak jenis perangkat.

Simulasi secara *real-time* perlu dilakukan antara dua perangkat dengan menggunakan protokol tertentu seperti TCP/IP.

Perlunya simulasi dari model jaringan yang berbeda untuk beberapa perangkat yang terhubung dalam satu jaringan.

Pembentukan *blockchain* yang lebih kompleks perlu dibuat untuk bisa memuat lebih banyak informasi.

LAMPIRAN

Fungsi dasar RSA [13]

```
import javax.crypto.Cipher;
plaintext = 'foobar';
cipher = Cipher.getInstance('RSA');
keygen                                  =
java.security.KeyPairGenerator.getInstanc
e('RSA');
keyPair = keygen.genKeyPair();
cipher.init(Cipher.ENCRYPT_MODE,
keyPair.getPrivate());
plaintextUnicodeVals                    =
uint16(plaintext);
plaintextBytes                          =
typecast(plaintextUnicodeVals, 'int8');
ciphertext                              =
cipher.doFinal(plaintextBytes)'
cipher.init(Cipher.DECRYPT_MODE,
keyPair.getPublic());
decryptedBytes                          =
cipher.doFinal(ciphertext);
decryptedText                           =
char(typecast(decryptedBytes, 'uint16'))'
```

Fungsi dasar pencarian nilai *hash*

```
data                                    =
'kvIf7#KHAP0#20180827T020745Z#A#12345.67#
1234.56#';
prev_hash                               =
'0000000000000000000000000000000000000000
0000000000000000000000000';
string = add_chara(prev_hash,data);
sha256hasher                            =
System.Security.Cryptography.SHA256Manage
d;
sha256                                  =
uint8(sha256hasher.ComputeHash(uint8(stri
ng)));
while sha256(1)~=0 || sha256(2)>=16
    string = add_chara(prev_hash,data);
    sha256                              =
uint8(sha256hasher.ComputeHash(uint8(stri
ng)));
end
hash = reshape(dec2hex(sha256)',64,1)';
```

Fungsi penambahan karakter acak pada program *hash*

```
function string = add_chara(prev,data)
symbols = ['A':'Z' '0':'9'];
stLength = 62-length(data);
nums = randi(numel(symbols),[1
stLength]);
st = symbols (nums);
prev_cut = prev(1:32);
string =
strcat(prev_cut,'#',data,'#',st);
end
```

Contoh penggunaan fungsi *tic* dan *toc* dalam fungsi *for* untuk pengukuran waktu komputasi secara berulang

```
t = zeros(1,100);
for n = 1:100
    A = rand(n,n);
    b = rand(n,1);
    tic;
    x = A\b;
    t(n) = toc;
end
plot(t)
```

DAFTAR PUSTAKA

[1] National Institute of Standards and Technology, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0," 2014.

[2] S. V. Nandury and B. A. Begum, "Big Data for Smart Grid Operation in Smart Cities," in *IEEE International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*, 2017.

[3] Laboratory for Advanced System Software, "Smart - UMass Trace Repository," 2018. [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart.

[4] The Research Perspective Ltd, "Smart Meter Electricity Trial Data Manifest," 2012.

[5] J. Gao *et al.*, "GridMonitoring: Secured Sovereign Blockchain Based Monitoring on Smart Grid," *IEEE Access*, vol. 4, pp. 2292–2303, 2018.

[6] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[7] W. Stallings, *Cryptography and Network Security: Principles and Practice*. Boston: Prentice Hal, Inc, 2011.

[8] D. Ariyus, *Pengantar Ilmu Kriptografi: Teori, Analisis, dan Implementasi*. Yogyakarta: Andi, 2008.

[9] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun. ACM*, vol. 21, pp. 120–126, 1978.

[10] E. Barker, *Recommendation for Key Management, Part 1: General, Revision 4*. Gaithersburg: Gaithersburg National Institute of Standards and Technology, 2016.

[11] W. Stallings, *Data and Computer Communications*, 10th ed. New Jersey: Prentice Hal, Inc, 2015.

[12] G. Walker, "Learn Me a Bitcoin - Target," *learnmeabitcoin.com*, 2016. [Online]. Available: http://learnmeabitcoin.com/glossary/target.

[13] A. Janke, "encryption - RSA code in matlab," *stackoverflow.com*, 2012. [Online]. Available: https://stackoverflow.com/a/9436658.