IMPLEMENTASI ALGORITMA ELGAMAL DAN FUNGSI HASH SHA-256 PADA DATA *ELECTRONIC VOTING* (*E-VOTING*)

SKRIPSI

OLEH:
INDRI FATIKHU AFLIKH
NIM. 18610036

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022

IMPLEMENTASI ALGORITMA ELGAMAL DAN FUNGSI HASH SHA-256 PADA DATA *ELECTRONIC VOTING* (*E-VOTING*)

SKRIPSI

OLEH:
INDRI FATIKHU AFLIKH
NIM. 18610036

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2022

IMPLEMENTASI ALGORITMA ELGAMAL DAN FUNGSI HASH SHA-256 PADA DATA *ELECTRONIC VOTING* (*E-VOTING*)

SKRIPSI

Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)

Oleh
INDRI FATIKHU AFLIKH
NIM. 18610036

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2022

IMPLEMENTASI ALGORITMA ELGAMAL DAN FUNGSI HASH SHA-256 PADA DATA *ELECTRONIC VOTING* (*E-VOTING*)

SKRIPSI

Oleh
Indri Fatikhu Aflikh
NIM. 18610036

Telah Diperiksa dan Disetujui Untuk Diuji

Malang, 26 Desember 2022

Dosen Pembimbing I	Dosen Pembimbing II
Hisyam Fahmi, M.Kom	Mohammad Nafie Jauhari, M.Si
NIP. 19890727 201903 1 018	NIDT. 19870218 20160801 1 056

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

IMPLEMENTASI ALGORITMA ELGAMAL DAN FUNGSI HASH SHA-256 PADA DATA *ELECTRONIC VOTING* (*E-VOTING*)

SKRIPSI

Oleh
Indri Fatikhu Aflikh
NIM. 18610036

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)
Tanggal 28 Desember 2022

Ketua Penguji	: Juhari, M.Si	.....
Anggota Penguji I	: Muhammad Khudzaifah, M.Si	.....
Anggota Penguji II	: Hisyam Fahmi, M.Kom	.....
Anggota Penguji III	: Mohammad Nafie Jauhari, M.Si	.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005

PERNYATAAN KEASLIAN TULISAN

Saya yang bertandatangan di bawah ini:

Nama : Indri Fatikhu Aflikh

NIM : 18610036

Program Studi : Matematika

Fakultas : Sains dan Teknologi

Judul Skripsi : Implementasi Algoritma ElGamal dan Fungsi Hash SHA-256 Pada Data *Electronic Voting* (*E-voting*)

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya sendiri. Bukan merupakan pengambilan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil tiruan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 28 Desember 2022

Yang membuat pernyataan,

Indri Fatikhu Aflikh

NIM. 18610036

MOTTO

Cukuplah Allah menjadi penolong kami dan Allah adalah sebaik-baik pelindung
(Q.S. *Ali-Imron*:173)

PERSEMBAHAN

Skripsi ini penulis persembahkan kepada seluruh pihak yang telah berperan bagi penulis sejak awal masa perkuliahan hingga terselesaikannya penelitian ini. Ucapan terima kasih penulis haturkan dengan segala ketulusan hati kepada kedua orang tua penulis, Suwito dan Ruchayati, serta kakak tercinta Aini Novita Amaliyah, yang senantiasa mengirimkan doa terbaiknya, mendidik, memberikan nasehat, motivasi, dukungan spiritual dan finansial serta kasih sayang yang tak terhingga kepada penulis.

Sahabat dan teman-teman penulis semasa perkuliahan, Clariza Adelina Rachma dan Candrani Sri Murtono, yang telah memberi semangat, motivasi, serta bantuan kepada penulis dalam menyelesaikan skripsi ini. Terima kasih telah membantu penulis dalam menyelesaikan perkuliahan serta segala tanggung jawab dalam organisasi dan komunitas.

Calon pasangan yang selalu memberikan semangat, motivasi, bantuan baik secara materiil maupun non materiil serta selalu mengingatkan agar segera terselesaikannya skripsi ini.

Diri saya sendiri yang mau berjuang dengan keras untuk dapat menuntaskan segala tanggung jawab perkuliahan dengan maksimal walaupun memiliki banyak kegiatan.

KATA PENGANTAR

Assalamu'alaikum Waarahmatullahi Wabarakatuh

Segala puji bagi Allah SWT yang telah melimpahkan seluruh rahmat, taufik serta hidayah-Nya, sehingga penulis masih diberi kesehatan dan kesempatan dalam menyelesaikan skripsi yang berjudul "Implementasi Algoritma ElGamal dan Fungsi Hash SHA-256 pada Data *Electronic Voting* (*E-Voting*)" sebagai salah satu persyaratan dalam memperoleh gelar sarjana Matematika. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita Nabi Muhammad SAW yang telah membawa kita dari zaman kegelapan menuju zaman yang terang benderang yakni agama Islam.

Dalam proses penyusunan skripsi ini, penulis banyak mendapatkan bimbingan, arahan, dan bantuan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. Sri Harini, M.Si., selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Elly Susanti, S.Pd., M.Sc, selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Hisyam Fahmi., M.Kom.,selaku dosen pembimbing I yang telah meluangkan waktunya untuk memberikan bimbingan, arahan, nasehat, solusi, dan banyak ilmu sehingga penulis dapat menyelesaikan proposal skripsi dengan baik.
5. Mohammad Nafie Jauhari, M.Si., selaku dosen pembimbing II yang telah memberikan bimbingan, arahan, dan banyak ilmu kepada penulis.
6. Juhari, M.Si, selaku Ketua Penguji dalam Seminar Proposal, Seminar Hasil, dan Sidang Skripsi yang telah memberikan kritik dan saran yang membangun kepada penulis.
7. Muhammad Khudzaifah, M.Si, selaku Anggota Penguji I yang telah memberikan kritik dan saran yang membangun serta berbagi ilmunya kepada penulis.

8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.
9. Orang tua tercinta, Ruchayati dan Suwito, kakak saya Aini Novita Amaliyah, keponakan saya Azfar dan Azmi, dan segenap keluarga yang tidak pernah berhenti dalam memberikan dukungan, semangat, motivasi, serta doa terbaik untuk kelancaran setiap urusan penulis.
10. Seluruh mahasiswa program studi Matematika angkatan 2018 yang telah memberikan bantuan materi, semangat, dan motivasi kepada penulis selama penyelesaian skripsi.

Penulis berharap semoga proposal skripsi ini dapat bermanfaat bagi pembaca maupun bagi penulis serta dapat dijadikan sebagai penambah wawasan ilmu matematika terutama dalam bidang komputasi.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, 28 Desember 2022

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

DAFTAR LAMPIRAN

.

ABSTRAK

Aflikh, Indri Fatikhu, 2022. **Implementasi Algoritma ElGamal dan Fungsi Hash SHA-256 pada data *Electronic Voting* (*E-Voting*)**. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Hisyam Fahmi, M.Kom., (II) Mohammad Nafie Jauhari, M.Si.

Kata Kunci: Algoritma ElGamal, Dekripsi, Enkripsi, Fungsi hash, SHA-256.

Terdapat dua jenis data, yakni data yang bersifat rahasia dan data yang bersifat tidak rahasia (terbuka). Data yang bersifat rahasia memerlukan keamanan yang kuat agar tidak mudah dilakukan modifikasi oleh pihak yang tidak berwenang. Pengamanan data dapat dilakukan dengan algoritma kriptografi. Penelitian ini menggunakan dua metode dalam melakukan pengamanan data diantaranya fungsi hash SHA-256 dan algoritma ElGamal. Adapun tujuan dari penelitian ini adalah untuk memberikan keamanan lebih pada data *electronic voting* yang dapat dimanipulasi oleh pihak tidak berwenang. Pada data pengguna yakni Nomor Induk Mahasiswa (NIM) akan dilakukan pengamanan menggunakan fungsi hash SHA-256. Pada fungsi hash SHA-256, seluruh perhitungan menggunakan bilangan biner yang memiliki simbol 0 dan 1. Penggunaan fungsi hash SHA-256 pada data NIM menghasilkan *message digest* dengan panjang 64 bit. Proses enkripsi dan dekripsi pada data pilihan yang diberikan oleh pengguna, peneliti menggunakan algoritma ElGamal dengan modifikasi kunci privat dan bilangan bulat acak. Sebelum melakukan proses enkripsi dan dekripsi, peneliti akan melakukan pembangkitan kunci menggunakan bilangan prima ribuan yang ditentukan secara acak. Penggunaan algoritma ElGamal memberikan hasil sepasang *ciphertext* (a_1, b_1) yang dapat memiliki ukuran dua kali dari ukuran pesan aslinya.

ABSTRACT

Aflikh, Indri Fatikhu, 2022. **On The Implementation of ElGamal Algorithm and SHA-256 Hash Function on Electronic Voting (E-Voting) Data**. Thesis. Mathematics Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisor: (I) Hisyam Fahmi, M.Kom., (II) Mohammad Nafie Jauhari, M.Si.

Keywords: ElGamal Algorithm, Encryption, Decryption, Hash Function, SHA-256.

There are two varieties of data, that is confidential data and non-confidential data. A confidential data requires strong security, which is rather not easy to modify by unauthorized parties. Data security can be done with cryptographic algorithms. This study used two methods in securing data including the SHA-256 hash function and the ElGamal algorithm. The purpose of this study is to provide more security to electronic voting data that can be manipulated by unauthorized parties. In user data, namely the Student Identification Number, a security will be carried out using the SHA-256 hash function. In the SHA-256 hash function, the entire calculation uses binary numbers that have symbols 0 and 1. The use of the SHA-256 hash function in Student Identification Number data results in a message digest with a length of 64 bits. The process of encryption and decryption on the selected data provided by the user, researchers used the ElGamal algorithm with the modification of the private key and random integers. Before carrying out the encryption and decryption process, the researcher will perform key generation using a randomly determined prime number of thousands. The use of the ElGamal algorithm gives the result of a pair of ciphertexts (a_1, b_1) that the size can be twice the size of the original message.

مستخلص البحث

أفلخ، إيندري فاتح. **٢٠٢٢. تطبيق خوارزمية الجمل ودالة التجزئة ٢٥٦-*SHA* على بيانات التصويت الإلكتروني (التصويت الإلكتروني)**. البحث العلمي. قسم الرّياضيات ، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف: (١) هشام فهمي، الماجستير، (٢) محمد نافع جوهري، الماجستير.

الكلمات المفتاحية: خوارزمية الجمل ، فك التشفير ، التشفير ، وظيفة التجزئة ، ٢٥٦-*SHA*.

تكون البيانات نوعين هما بيانات سرية وغير سرية (المفتوحة). تتطلب البيانات السرية أمانا قويا ، وليس من السهل تعديله من قبل أطراف غير مصرح لها. يمكن إجراء أمان البيانات باستخدام خوارزميات التشفير . استخدمت هذه الدراسة طريقتين في تأمين البيانات بما في ذلك دالة التجزئة ٢٥٦-*SHA*.وخوارزمية الجمال. الغرض من هذه الدراسة هو توفير المزيد من الأمان لبيانات التصويت الإلكترونية التي يمكن التلاعب بها من قبل أطراف غير مصرح لها. في بيانات المستخدم ، أي رقم تعريف الطالب، سيتم تنفيذ الأمان باستخدام وظيفة التجزئة ٢٥٦-*SHA*. في دالة التجزئة ٢٥٦ -*SHA* ، يستخدم الحساب بأكمله أرقاما ثنائية لها رمزان 0 و 1. ينتج عن استخدام دالة التجزئة ٢٥٦ -*SHA*. في بيانات رقم تعريف الطالب ملخص رسالة بطول ٦٤ بت. عملية التشفير وفك التشفير على البيانات المختارة المقدمة من المستخدم، استخدم الباحثون خوارزمية الجمل مع تعديل المفتاح الخاص والأعداد الصحيحة العشوائية. قبل تنفيذ عملية التشفير وفك التشفير، سيقوم الباحثون بإنشاء المفاتيح باستخدام عدد أولي محدد عشوائيا من الآلاف. يعطي استخدام خوارزمي *ElGamal* نتيجة زوج من النصوص المشفرة (a_1, b_1) التي يمكن أن يكون حجمها ضعف حجم الرسالة الأصلية.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Masalah keamanan dan kerahasiaan data atau informasi pada saat ini terutama bagi suatu organisasi atau perusahaan tentu sangat penting. Data terbagi ke dalam dua jenis, yakni data yang bersifat rahasia dan data yang bersifat tidak rahasia. Data yang bersifat rahasia akan sangat dijaga, diperhatikan serta tidak mudah digandakan oleh orang lain. Sedangkan data tidak rahasia biasanya tidak terlalu diperhatikan dan orang dapat melakukan penggandaan dengan mudah. Adanya proses penggandaan data yang bersifat rahasia menjadikan kita untuk melakukan pengamanan lebih pada data guna menjamin kerahasiaannya. Keamanan yang digunakan dapat berupa kunci untuk membuka data terkait, dimana kunci yang digunakan hanya diketahui oleh pengirim dan penerima pesan. Proses mengubah data yang dapat dibaca menjadi data yang tidak dapat dibaca disebut dengan enkripsi, sedangkan proses mengembalikan data yang tidak dapat dibaca menjadi data semula yaitu dekripsi.

Pemungutan suara atau sering kita sebut dengan *voting* tidak hanya dilakukan pada tingkat negara, tetapi juga dalam tingkat masyarakat kecil seperti kampus. *Voting* digunakan untuk mengumpulkan aspirasi mahasiswa dan kemudian menemukan hasil yang dianggap terbaik untuk memecahkan masalah (Arifin & Sajono, 2013). Pemilihan umum yang dilaksanakan secara langsung memiliki beberapa kendala seperti adanya keterlambatan pembagian kartu pemilih dan adanya kartu pemilih yang cacat sehingga menyebabkan berkurangnya kartu

pemilih pada hari pemilihan. Permasalahan lainnya juga dapat ditimbulkan oleh pengamanan kotak suara yang tidak diberikan segel ataupun rusak sehingga mengakibatkan sengketa antara Pasangan Calon karena ditengarai terjadi manipulasi surat suara (Ardilla & Asrinaldi, 2019). Kegiatan *voting* yang tidak dilaksanakan secara bersamaan dapat menimbulkan beberapa permasalahan diantaranya waktu pemilihan serta proses perhitungan yang membutuhkan banyak waktu dan terjadinya kecurangan. Persoalan tersebut dapat memicu ikatan yang tidak seimbang serta adanya kemungkinan terjadi konflik antar lingkup masyarakat. Oleh karena itu dibutuhkan strategi yang dapat melindungi kerahasiaan dan keaslian dari hasil pelaksanaan *voting* (Arifin & Sajono, 2013). Masalah yang timbul dalam pemilihan umum dapat ditekan dengan penggunaan teknologi yang menjadikan pemilu dapat berjalan dengan jujur dan dapat mengurangi masalah yang terjadi. Perkembangan teknologi informasi saat ini telah membawa perubahan besar pada masyarakat, seperti bagaimana pemungutan suara dilakukan. Penggunaan teknologi komputer dalam melakukan pemungutan suara dikenal dengan *electronic voting*, atau lebih umum disebut *e-voting*.

Penggunaan teknologi tersebut dapat membantu mempercepat proses pemungutan dan perhitungan suara serta dapat mengurangi risiko kekeliruan dan mengurangi pengeluaran biaya (Suganda, 2019). Salah satu penerapan enkripsi dan dekripsi pesan dapat dilakukan pada sistem pemungutan suara berbasis elektronik. Terdapat beberapa variasi dari implementasi *e-voting*, seperti penggunaan internet sebagai skema pemungutan suara, penggunaan *touch screen* sebagai pengganti kertas suara, penggunaan *smart card* untuk otentikasi pemilih, dan lain sebagainya (Arifin & Sajono, 2013).

Keamanan yang kuat dibutuhkan pada sistem *e-voting* karena terdapat banyak sela keamanan pada sistem digital yang dapat digunakan untuk melakukan pencurian data. Salah satu teknik pengamanan data yang dapat dilakukan yakni dengan menerapkan algoritma ElGamal untuk mengenkripsi hasil *voting* yang dikirimkan oleh pemilih. Algoritma ElGamal merupakan salah satu jenis dari algoritma asimetris. Alasan peneliti menggunakan algoritma ElGamal dikarenakan sulitnya perhitungan logaritma diskrit yang digunakan sehingga kecil kemungkinannya untuk dilakukan manipulasi data. Algoritma ElGamal dapat digunankan untuk enkripsi dan dekripsi suatu pesan. Pada penelitian ini, hasil *voting* dari pemilih yang bersifat rahasia menjadikan perlu adanya proses enkripsi hasil *voting* tersebut. Beberapa unsur yang dibutuhkan pada algoritma ElGamal diantaranya bilangan prima (p), akar primitif dari bilangan prima (g), kunci privat (x), bilangan bulat (k), serta pesan yang dikodekan ke dalam bentuk ASCII. Selanjutnya dapat dilakukan pembangkitan kunci, enkripsi, dan dekripsi pesan dengan algoritma ElGamal.

Fungsi hash satu arah (*one way hash function*) merupakan algoritma yang dapat memroses pesan untuk menghasilkan gambaran ringkas yang disebut *message digest*. Algoritma ini memungkinkan untuk menentukan integritas pesan dimana adanya perubahan pesan dengan kemungkinan yang sangat tinggi akan menghasilkan *message digest* yang berbeda. Salah satu macam dari fungsi hash satu arah adalah SHA-256 yang memiliki ukuran blok 512 bit. SHA-256 menggunakan enam logika yang merupakan kombinasi dasar seperti AND, OR, XOR, pergeseran bit kekanan (*right shift*), dan rotasi kekanan (*right rotate*).

Pada penelitian sebelumnya yang telah dilakukan oleh Luthfiatun Nisa, Tutuk Indriyani, dan Maretha Ruswiansari (Nisa dkk, 2020) membahas proses enkripsi dan dekripsi citra dan teks menggunakan algoritma ElGamal dan dikombinasikan dengan algoritma Diffie-Hellman. Pada penelitian tersebut ditambahkan algoritma Diffie-Hellman sebagai proses pertukaran kunci. Selanjutnya proses pembangkitan kunci, enkripsi, dan dekripsi pesan menggunakan algoritma ElGamal. Pada penelitian Irena Kusuma Dewi, Rahma Intari, dkk (Dewi, 2021) membahas proses enkripsi dan dekripsi menggunakan algoritma ElGamal. Penelitian tersebut melakukan proses pembangkitan kunci, enkripsi, dan dekripsi menggunakan bantuan suatu program buatan. Perbedaan panjang pesan yang dimasukkan mempengaruhi adanya perbedaan waktu pada saat proses enkripsi dan dekripsi. Namun, dengan perbedaan itu tidak berpengaruh pada output pesan karena hasil yang didapatkan sama dengan masukannya. Penelitian yang dilakukan oleh Santi Sulastri dan Riana Defi Mahadji Putri (Sulastri & Putri, 2018) membahas proses penyandian password pada saat halaman login. Password yang diinputkan akan dilakukan *genarate* menggunakan MD5 dan disimpan dalam sebuah variabel yang diberi nama "atr". Hasil dari tahap pertama akan diambil karakter pertama dan juga terakhir untuk ditambahkan sebagai *padding* pada password yang diinputkan oleh pengguna.

Sejak dulu Agama Islam mengajarkan dalam ajarannya tentang esensi dalam menjaga amanah. Dikarenakan amanah merupakan suatu perkara yang dititipkan kepada orang lain untuk dijaga dan dipertanggung jawabkan. Telah dijelaskan dalam Al-Qur'an terkait anjuran untuk bersikap amanah, yaitu terdapat di dalam surat Al-Mukminum ayat 8 – 11 yang berbunyi:

"Dan (sungguh beruntung) orang yang memelihara amanat-amanat dan janjinya, serta orang yang memelihara shalatnya. Mereka itulah orang yang akan mewarisi, (yakni) yang akan mewarisi (surga) Firdaus. Mereka kekal di dalamnya."

Apabila seseorang telah bersedia untuk menjaga suatu amanah, secara tidak langsung orang tersebut telah berjanji akan amanah tersebut. Maka untuk melaksanakan anjuran tersebut dalam proses perpindahan data atau informasi dibutuhkan jalan keluar untuk menjaga amanah yang diberikan tetap terjaga dan dapat diterima oleh pihak yang dituju.

Berdasarkan penjabaran tersebut, pada penelitian ini penulis akan melakukan hashing pada identitas pengguna atau pemilih dengan fungsi hash SHA-256 sebelum melakukan pemilihan. Hasil enkripsi pada fungsi hash tersebut akan diautentikasi dengan data pemilih yang terdaftar pada database. Pemilih akan diberikan izin untuk mengirimkan pilihannya apabila pemilih tersebut terdaftar pada database dan belum melakukan pemilihan. Sedangkan untuk hasil *voting* yang dikirimkan oleh pemilih akan dilakukan enkripsi menggunakan algoritma ElGamal. Hasil *voting* yang dikirimkan oleh pemilih akan digabungkan dengan kode user dan kemudian akan disimpan pada database. Dekripsi pada algoritma ElGamal akan dilakukan setelah semua pemilih telah mengirimkan pilihannya untuk mengetahui hasil dari pemilihan. Adanya autentukasi pengguna menggunakan algoritma SHA-256 serta proses enkripsi hasil enkripsi menggunakan algoritma ElGamal diharapkan dapat mengurangi tingkat kecurangan pada saat pengiriman *voting* palsu atau *voting* yang tidak dilakukan oleh pengguna yang semestinya.

1.2 Rumusan Masalah

Berdasarkan penjabaran yang telah dituliskan pada latar belakang, rumusan masalah pada penelitian ini adalah bagaimana implementasi algoritma ElGamal dan fungsi hash SHA-256 pada data *e-voting*.

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini berdasarkan rumusan masalah adalah untuk mengetahui hasil dari implementasi algoritma ElGamal dan fungsi hash SHA-256 pada proses pengamanan data *e-voting*.

1.4 Manfaat Penelitian

Beberapa manfaat yang didapatkan dari penelitian ini yaitu:

1. Adanya penghematan anggaran yang digunakan untuk membiayai proses pemilihan umum.
2. Adanya penghematan waktu pada saat melakukan pemilihan.
3. Terhindar dari adanya hasil *voting* palsu atau tidak sah yang disebabkan oleh penyediaan kertas suara palsu.
4. Meningkatkan keamanan data pengguna dalam proses pemungutan suara.
5. Menambah pengetahuan bagi peneliti serta pembaca dalam bidang kriptografi.

1.5 Batasan Masalah

Beberapa batasan masalah diberikan oleh penulis agar pembahasan penelitian tidak meluas, diantaranya:

1. Data pengguna pada penelitian ini berupa Nomor Induk Mahasiswa (NIM) yang memiliki 8 karakter.

2. Karakter teks yang digunakan dalam penelitian ini akan dikodekan kedalam bentuk karakter ASCII.
3. Panjang karakter pada setiap data terbatas dengan jumlah maksimal 50 karakter.
4. Proses *hashing* menggunakan fungsi hash SHA-256 dilakukan pada Nomor Induk Mahasiswa (NIM).
5. Bilangan prima yang digunakan pada algoritma ElGamal antara 1000 – 5000.
6. Akar primitif (g) menggunakan akar primitif terkecil dari bilangan prima yang digunakan.
7. Kunci privat (x) mengambil angka ketiga dan digabungkan dengan jumlah dua angka terakhir dari NIM.
8. Bilangan integer (k) merupakan pasangan dari jumlah dua angka terakhir NIM dipasangkan dengan bilangan ASCII dari *vote* yang dikirimkan.

1.6 Definisi Istilah

1. Algoritma : Urutan tahapan yang terurut secara sistematis dan logis untuk menyelesaikan sebuah permasalahan
2. Bit : Digit biner yang memiliki nilai 0 atau 1
3. *Byte* : Sekelompok delapan bit yang diperlakukan sebagai entitas tunggal
4. *Ciphertext* : Merupakan teks tersandi yang berasal dari proses enkripsi
5. Dekripsi : Prosedur mengubah pesan yang tidak terbaca menjadi pesan yang dapat dibaca oleh manusia

6.	ElGamal	:	Algoritma kriptografi kunci publik yang dibuat oleh Taher ElGamal
7.	Enkripsi	:	Proses pengkodean suatu pesan terbaca menjadi pesan yang tidak terbaca oleh pengguna
8.	*Hash*	:	Suatu kode yang diperoleh dari hasil enkripsi. Umumnya terdiri dari huruf maupun angka yang acak
9.	*Input*	:	Data yang dimasukkan untuk di proses
10.	*Output*	:	Hasil yang dikeluarkan setelah dilakukannya proses dari pesan masukan (*input*)
11.	*Primitive root*	:	Penyelesaian kongruensi non-linear yang terkait dengan perpangkatan bilangan prima
12.	*SHA*	:	*Secure Hash Algorithm*. Algoritma kriptografi yang merupakan salah satu jenis hash dan dibuat oleh *National Institute of Standard and Technology* (*NIST*).
13.	*Word*	:	Sekelompok 32 bit (4 byte) atau 64 bit (8 byte), tergantung pada algoritma hash yang aman
14.	*Voting*	:	Pemungutan suara

BAB II

KAJIAN TEORI

2.1 Sistem Bilangan

Sistem bilangan adalah sistem penulisan untuk merepresentasikan suatu bilangan. Sistem bilangan yang paling umum digunakan oleh manusia adalah bilangan desimal yang terdiri dari 10 (0 sampai dengan 9) simbol untuk mewakili suatu besaran. Sistem bilangan desimal disebut juga dengan sistem bilangan berbasis 10 yang ditulis dalam ukuran yang diperkecil (*subscrib*) seperti contohnya 100_{10}, akan tetapi biasanya pada sistem bilangan desimal tidak dituliskan basisnya.

Sistem bilangan yang digunakan dalam komputer terdiri dari beberapa jenis diantaranya:

1. Sistem Bilangan Biner

   Sistem bilangan yang menggunakan dua simbol (0 dan 1) disebut juga dengan sistem bilangan berbasis 2. Biner merupakan bilangan dasar yang digunakan dalam sistem komputer digital. Penulisan bilangan biner biasanya dikelompokkan per 4 bilangan, misalnya: 1010 0001. Sistem bilangan dengan basis pada bilangan biner menunjukkan eksponen dengan basis 2, yaitu 2^0=1, 2^1=2, 2^2=4, dan seterusnya. Pada bilangan biner, setiap digitnya disebut dengan bit. Bit paling kanan disebut *least significant bit* (LSB) dan bit paling kiri disebut dengan *most significant bit* (MSB).

   Contoh:

   $0010_2 = 0010 = 2_{10}$

   $1010_2 = 1010 = 10_{10}$

2. Sistem Bilangan Oktal

Bilangan oktal merupakan bilangan dengan basis 8 dan memiliki delapan simbol bilangan berbeda yaitu 0,1,2,3,4,5,6, dan 7. Nilai tempat sistem bilangan oktal merupakan perpangkatan dan nilai 8 sebagai berikut.

Tabel 2.1 Bilangan Oktal

Posisi Digit (Dari Kanan)	Nilai Tempat
1	$8^0=1$
2	$8^1=8$
3	$8^2=64$
4	$8^3=512$

3. Sistem Bilangan Desimal

Bilangan desimal adalah bilangan dengan basis 10 dan memiliki 10 simbol yaitu 0 hingga 9. Setiap tempat memiliki nilai kelipatan dari 10^0, 10^1, 10^2, dan seterusnya. Setiap tempat memiliki besaran tertentu yang secara urut dimulai dari kanan disebut dengan satuan, ratusan, ribuan, dan seterusnya.

Contoh:

Angka Desimal 10843 (10843_{10})

Tabel 2.2 Penyelesaian Angka Desimal

Urutan	Penjabaran	Perhitungan	Hasil
Pertama	$3 . 10^0$	3 . 1	2
Kedua	$4 . 10^1$	4 . 10	40
Ketiga	$8 . 10^2$	8 . 100	800
Keempat	$0 . 10^3$	0 . 1000	0

Urutan	Penjabaran	Perhitungan	Hasil
Kelima	$1 . 10^4$	1 . 10000	10000
Jumlah			10843

4. Sistem Bilangan Heksadesimal

Sistem bilangan heksadesimal memiliki basis 16 dan menggunakan 16 macam simbol, yaitu 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F. Pada sistem bilangan heksadesimal, A=10, B=11, C=12, D=13, E=14, dan F=15. Nilai tempat pada bilangan heksadesimal merupakan perpangkatan dari 16.

Contoh:

$D8_{(16)} = \ldots_{(10)}$

Tabel 2.3 Penyelesaian Bilangan Desimal

Urutan	Simbol	Penjabaran	Hasil
Pertama	8	$8 . 16^0$	8
Kedua	D	$13 . 16^1$	208
Jumlah			216

Sistem bilangan tersebut tidak hanya dipergunakan pada letaknya masing-masing. Sistem bilangan juga dapat dilakukan konversi seperti sistem bilangan biner yang dikonversikan kedalam sistem bilangan desimal. Teknis konversi pada setiap bilangan dijelaskan sebagai berikut.

1. Konversi Bilangan Biner ke Desimal

Konversikan bilangan biner 01111010_2 ke dalam bentuk desimal.

$$01111010_2 = 0 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$01111010_2 = 0 \times 128 + 1 \times 64 + 1 \times 32 + 1 \times 16 + 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 0$$

$$01111010_2 = 64 + 32 + 16 + 8 + 2$$

$$01111010_2 = 122_{10}$$

Berdasarkan ilustrasi di atas, maka bilangan biner desimal dari 01111010_2 adalah 122_{10}.

2. Konversi Bilangan Desimal ke Biner

Cara untuk mengkonversi bilangan desimal kedalam bilangan biner adalah dengan pembagian. Bilangan desimal yang akan dikonversi secara berturut-turut dibagi dengan 2, dengan memperhatikan sisa pembagiannya. Sisa pembagian akan bernilai 0 atau 1. Bilangan 0 atau 1 tersebut akan menjadi susunan bilangan biner. Sebagai contoh, akan dilakukan konversi dari bilangan desimal 122 kedalam bentuk biner.

$\frac{122}{2}$ = 61 sisa 0 *Least Significant Bit* (LSB)

$\frac{61}{2}$ = 30 sisa 1

$\frac{30}{2}$ = 15 sisa 0

$\frac{15}{2}$ = 7 sisa 1

$\frac{7}{2}$ = 3 sisa 1

$\frac{3}{2}$ = 1 sisa 1

$\frac{1}{2}$ = 0 sisa 1

$\frac{0}{2}$ = 0 sisa 0 *Most Significant Bit* (MSB)

Sehingga, bermula pada MSB menuju LSB, bilangan desimal 122_{10} yang dikonversi ke dalam biner menjadi 01111010.

3. Konversi Bilangan Desimal ke Heksadesimal

Bilangan heksadesimal didapatkan dengan basis 16 akan membagi bilangan desimal yang akan dikonversi. Berikut cara melakukan konversi bilangan desimal ke bilangan heksadesimal. Bilangan desimal = 10846, apabila dikonversi kedalam bentuk bilangan heksadesimal adalah sebagai berikut:

10846 : 16 = 677 sisa 14, 14 = E

677 : 16 = 42 sisa 5, 5 = 5

42 : 16 = 2 sisa 10, 10 = A

2 : 16 = 0 sisa 2, 2 = 2

Maka, bilangan desimal 10846 dalam heksadesimal adalah 2A5E.

4. Konversi Bilangan Heksadesimal ke Desimal

Bilangan heksadesimal 2A5E, apabila dikonversi ke bilangan desimal akan dikalikan dengan basis perpangkatan 16.

2A5E :

2 = $2 . 16^3$ = 2 . 4096 = 8192

A = $10 . 16^2$ = 2 . 256 = 2560

5 = $5 . 16^1$ = 5 . 16 = 80

E = $14 . 16^0$ = 14 . 1 = 14

Jumlah = 8192 + 2560 + 80 + 14 = 10846

2.2 Operator Bitwise

Operasi bitwise adalah operator yang bekerja pada manipulasi bit. Operator bitwise hanya dapat dilakukan pada operand dengan tipe char (*character*) dan int (*integer*). Operator bitwise yang akan digunakan dalam penulisan ini adalah sebagai berikut.

1. Operasi "Dan" ($\wedge$)

   Operasi "dan" melakukan perbandingan pada dua ekspresi dan menjelaskan apakah dua ekspresi tersebut adalah benar. Pada operasi ini angka "1" berlaku sebagai "benar", sedangkan angka "0" berlaku sebagai salah. Apabila dua bit yang dibandingkan adalah 1, maka hasilnya adalah 1 ($1 \times 1 = 1$). Sebaliknya, jika dua bit yang dibandingkan adalah 0, maka hasilnya adalah 0 ($1 \times 0 = 0, 0 \times 1 = 0, 0 \times 0 = 0$).

2. Operasi "Atau" ($\vee$)

   Operasi "atau" membandingkan dua ekspresi dan menjelaskan apakah setidaknya salah satu dari keduanya adalah benar. Pada operasi "atau" berlaku: $1|1 = 1$, $1|0 = 1$, $0|1 = 1$, $0|0 = 0$.

3. Operasi "Not" ($\neg$)

   Operasi "not" merupakan operasi negasi atau membalik bit yang diberikan. Apabila bit tersebut adalah "1", maka akan diganti ke "0". Sebaliknya, apabila bit yang diberikan adalah "0", maka akan diganti ke "1".

4. Operasi "XOR" ($\oplus$)

   Operasi "xor" memiliki arti *exclusive or* yang berarti apabila bit yang diberikan keduanya adalah sama akan menghasilkan angka 0.

Tabel 2.4 Operasi XOR

X	**Y**	**X⊕Y**
1	1	0
1	0	1
0	0	0
0	1	1

5. Operasi Penjumlahan (+)

Operasi penjumlahan pada bilangan biner seperti halnya pada bilangan desimal. Penjumlahan pada bilangan biner akan memperhatikan dua unsur yakni jumlah (sum) dan sisa (*carry out*). Dua buah biner yang dijumlahkan apabila memiliki sisa akan ditambahkan pada penjumlahan dua bilangan biner di depannya. Aturan yang berlaku pada penjulahan biner adalah sebagai berikut.

$0 + 0 = 0$

$0 + 1 = 1$

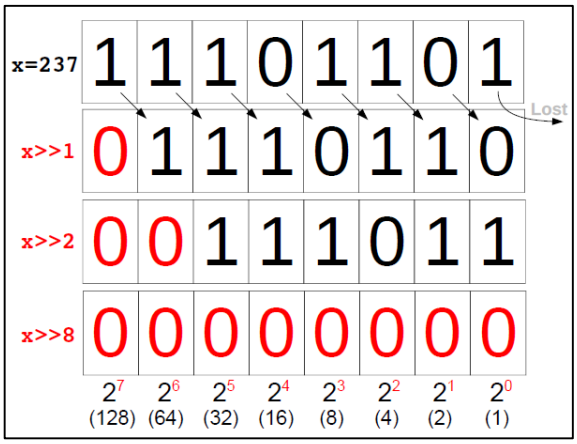
$1 + 0 = 1$

$1 + 1 = 0, +1$ sebagai *carry*

$1 + 1 + 1 = 1, +1$ sebagai *carry*

6. *Shift Right*

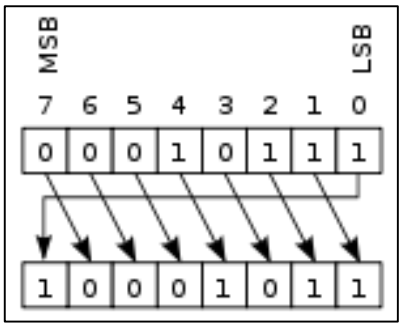
Operasi *shift right* melakukan pergeseran sejumlah bit ke kanan atau melakukan perpindahan arah bit ke kanan. Sebagai contoh, terdapat bilangan desimal $x = 237$ akan dilakukan *shift right* sebesar 2 mendapatkan hasil sebagai berikut:



Gambar 2. 1 Operasi Shift Right

7. *Rotate Right* (ROTR(x))

Rotasi (atau pergeseran melingkar) adalah operasi yang mirip dengan pergeseran kecuali bahwa bit yang berada di satu ujung diletakkan kembali ke ujung lainnya. Dalam rotasi kiri, bit yang berada di ujung kiri diletakkan kembali di ujung kanan. Dalam rotasi kanan, bit yang berada di ujung kanan diletakkan kembali di ujung kiri.



Gambar 2. 2 Rotate Right

2.3 Kriptografi

Kriptografi merupakan ilmu yang berhubungan dengan teknik enkripsi dimana akan dilakukan pengacakan data menggunakan suatu kunci enkripsi menjadi sesuatu yang tidak dapat dibaca oleh pihak yang tidak seharusnya. Dekripsi yang dilakukan dengan memanfaatkan kunci dekripsi akan menghasilkan data asli. Proses enkripsi dilakukan menggunakan suatu algoritma dengan beberapa

parameter. Rahasia terletak di beberapa parameter yang digunakan, sehingga kunci ditentukan oleh parameter (Kromodimoeljo, 2009).

Salah satu ilmu matematika yang memiliki keterkaitan dengan bidang keamanan informasi seperti integritas data, keaslian entitas, dan keaslian data adalah kriptografi. Berbagai macam metode dalam upaya untuk mengamankan data digunakan pada kriptografi. Keamanan merupakan bentuk kegiatan untuk menjaga sesuatu hal dari berbagai jenis ancaman dan gangguan (Shita & Hin, 2018).

Menurut Purwadi (2014) dalam karya tulis Rizky Tahara Shita dan Lauw Li Hin (2018), algoritma kriptografi terdiri dari 3 fungsi dasar, yaitu:

1. Enkripsi merupakan proses pengubahan pesan asli (*plaintext*) yang dikirimkan menjadi kode yang tidak dimengerti agar terjaga kerahasiaannya serta menggunakan algoritma pengkodean.
2. Dekripsi merupakan proses pengubahan pesan dalam bentuk kode (*ciphertext*) menjadi bentuk pesan asli (*plaintext*) dengan algoritma yang berbeda dari enkripsi.
3. Kunci yang dimaksud adalah kunci yang digunakan dalam proses enkripsi dan dekripsi.

Teknik enkripsi simetris digunakan pada kriptografi klasik dimana kunci dekripsi sama dengan kunci enkripsi. Sedangkan teknik enkripsi asimetris digunakan pada *public key cryptography* dimana kunci dekripsi tidak sama dengan kunci enkripsi. Diperlukan komputasi yang lebih intensif pada teknik enkripsi asimetris dibandingkan dengan teknik enkripsi simetris, dikarenakan adanya penggunaan bilangan-bilangan yang sangat besar pada enkripsi asimetris. Kriptografi asimetris lebih tinggi dari pada kriptografi simetris, tetapi kriptografi

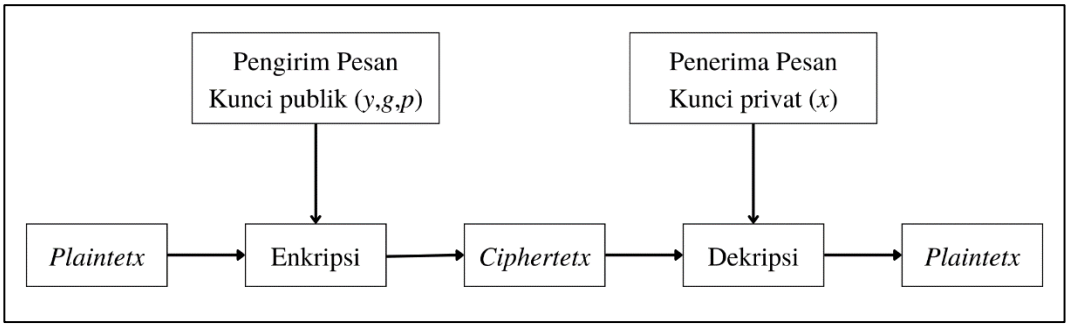
kunci publik sangat berfungsi untuk manajemen kunci dan tanda tangan digital (Kromodimoeljo, 2009).

2.4 Algoritma ElGamal

Algoritma ElGamal diciptakan pada tahun 1984 oleh Taher ElGamal. Algoritma ElGamal banyak dimanfaatkan sebagai tanda tangan digital, kemudian dilakukan perubahan sehingga dapat dimanfaatkan pada enkripsi dan dekripsi. Sulitnya perhitungan logaritma diskrit merupakan kekuatan kriptografi ElGamal sehingga dimanfaatkan dalam perangkat lunak keamanan yang dikembangkan oleh GNU, program PGP (*Pretty Good Privacy*), dan pada sistem keamanan lainnya (Fajrin dkk, 2019).

Kekurangan dari algoritma ElGamal terdapat pada pembangkitan kunci yang menggunakan logaritma diskrit dan metode enkripsi. Diperlukan komputasi yang besar pada proses pembangkitan kunci sehingga hasil enkripsi memiliki ukuran dua kali dari ukuran semula (*plaintext*). Maka dari itu, dibutuhkan sumber daya yang baik dan *processor* yang mampu melakukan komputasi yang besar (Purnomo, 2018).

Terdapat tiga tahapan yang akan dilakukan dalam penggunaan algoritma ElGamal, yaitu: proses pembangkitan kunci, proses enkripsi, dan proses dekripsi.



Gambar 2.3 Proses Enkripsi dan Dekripsi algoritma ElGamal

2.4.1 Pembangkitan Kunci

Algoritma ElGamal membutuhkan sepasang kunci yang dibangkitkan dengan menentukan bilangan prima p dan dua buah bilangan acak, yaitu g dan x dengan syarat $g < p$ dan $1 \leq x \leq p - 2$ yang memenuhi persamaan

$$y = g^x \ mod \ p. \tag{2.1}$$

Keterangan:

y = Kunci publik

g = Generator (akar primitif dari bilangan prima)

p = Bilangan prima

Berdasarkan persamaan tersebut nilai (y, g, p) merupakan sepasang kunci publik, sedangkan $(x,\ p)$ merupakan sepasang kunci pribadi (Kurniadi, 2015). Properti yang dimanfaatkan dalam pembangkitan kunci algoritma ElGamal sebagai berikut:

a. Bilangan prima p bersifat tidak rahasia
b. Bilangan acak g (generator, $g < p$) didapatkan dari perhitungan akar primitif bilangan prima serta bersifat tidak rahasia
c. Bilangan acak x yang akan menjadi kunci pribadi (private key, $x < p$) bersifat rahasia
d. $y = g^x \ mod \ p$ menjadi pasangan kunci publik (public key) bersama g dan p (y, g, p) bersifat tidak rahasia

2.4.2 Proses Enkripsi

Proses enkripsi memanfaatkan kunci publik (y, g, p) dan bilangan acak k dengan syarat $1 \leq k \leq p - 2$. Setiap karakter dalam pesan dilakukan enkripsi dengan bilangan k yang berbeda untuk meningkatkan keamanan dari *ciphertext*

yang dihasilkan. Satu karakter m yang dituliskan ulang dengan menggunakan bilangan bulat ASCII dienkripsi menggunakan rumus sebagai berikut.

$$a = g^k \ mod \ p \tag{2.2}$$

$$b = m \ y^k \ mod \ p \tag{2.3}$$

Keterangan:

g = Generator (akar primitif dari bilangan prima)

m = *Plaintext*

p = Bilangan prima

Proses enkripsi pada algoritma ElGamal akan menghasilkan kode dalam bentuk blok yang terdiri atas dua nilai (a, b) (Purnomo, 2018). Hasil enkripsi pesan disusun secara berurutan menjadi $a_1,\ b_1,\ a_2,\ b_2, \dots, a_n,\ b_n$.

2.4.3 Proses Dekripsi

Pada proses dekripsi digunakan sepasang kunci pribadi (x, p) untuk mendekripsi (a, b) menjadi *plaintext* dengan persamaan:

$$m = b.c \ mod \ p. \tag{2.4}$$

Nilai dari variabel c dapat dicari dengan menggunakan persamaan sebagai berikut:

$$c = a^{p-1-x} \ mod \ p. \tag{2.5}$$

Keterangan:

m = *Plaintext*

b = *Chipertext*

p = Bilangan prima

2.5 Fungsi *Hash*

Fungsi hash merupakan kriptografi primitif yang penting dan banyak digunakan dalam protokol. Mereka melakukan perhitungan pada pokok pesan yang merupakan bit string berukuran pendek dengan panjang tetap. Beberapa pesan tertentu, pokok pesan atau nilai hash, dapat dilihat sebagai sidik jari dari sebuah pesan, yaitu representasi unik dari sebuah pesan. Pada saat akan menggunakan fungsi hash tidak dibutuhkan suatu kunci. Fungsi hash juga banyak digunakan untuk aplikasi kriptografi lainnya, misalnya untuk menyimpan hash kata sandi atau derivasi kunci.

Fungsi hash yang tidak memiliki kunci dalam proses enkripsi sehingga dibutuhkan beberapa properti khusus agar fungsi hash menjadi lebih aman. Seperti yang terjadi dalam kriptografi, mudah juga terdapat suatu serangan menggunakan kelemahan fungsi hash. Beberapa properti yang perlu dimiliki fungsi hash agar tetap aman, diantaranya:

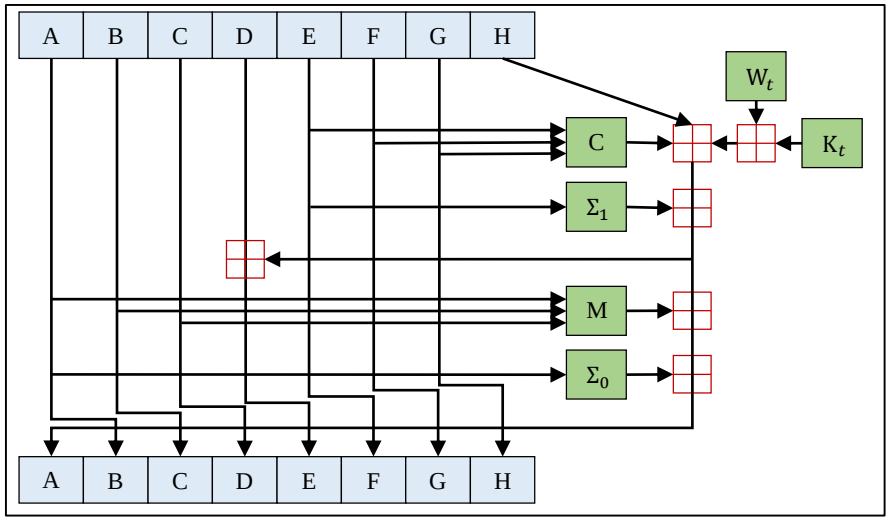
1. Ukuran pesan arbitrer $h(x)$ dapat diimplementasikan pada pesan x dengan ukuran yang tidak ditentukan.
2. Panjang output fungsi hash yang tetap $h(x)$ menghasilkan nilai hash z dengan panjang tetap.
3. Efisiensi $h(x)$ relatif mudah dilakukan perhitungan.
4. Resistensi *preimage* untuk keluaran z yang diberikan, tidak mungkin untuk mencari input x lainnya sedemikian rupa sehingga $h(x) = z$, $h(x)$ adalah satu-arah

5. Resistensi *preimage* kedua mengingat x_1 dan dengan demikian $h(x_1)$, secara komputasi tidak mungkin untuk mendapatkan x_2 sedemikian rupa sehingga $h(x_1) = h(x_2)$
6. Pada resistensi tabrakan, secara komputasi tidak mungkin untuk menemukan pasangan $x_1 = x_2$ sedemikian rupa sehingga $h(x_1) = h(x_2)$

2.6 Algoritma SHA-256

Pada tahun 1993, SHA yang termasuk bagian dari MD fungsi hash, dipublikasikan oleh *American National Institute for Standard and Technology* (NIST) serta dijadikan sebagai standar FIPS. Model SHA pertama kali adalah SHA-0 yang memiliki sedikit perubahan pada tahun 1994 kemudian hadir sebagai SHA-1. Fungsi SHA pada tahun 2000 diperkenalkan dengan ukuran pesan digest 256 bit kemudian dikenal dengan SHA-256, 384 bit dikenal dengan SHA-384, dan 512 bit dikenal dengan SHA-512 serta kemudian dijadikan sebagai standar FIPS pada tahun 2022 .

Perhitungan nilai *messages digest* dari sebuah pesan yang memiliki panjang maksimum 2^{64} bit memanfaatkan algoritma SHA-256. Diperlukan sebuah *message schedule* yang terdiri dari 64 elemen 32 bit kata, delapan buah variabel 32 bit, dan variabel penyimpanan nilai *hash* delapan buah kata 32 bit (Sebastian, 2007). SHA-256 melakukan konversi pesan sebagai masukan ke bentuk *message digest* 256 bit. Pesan masukan yang memiliki panjang lebih pendek dari 2^{64} bit, harus di proses oleh 512 bit dalam satuan, kemudian akan menjadi sebuah message digest 256-bit.



Gambar 2.4 Alur Satu Kali Iterasi Fungsi Hash SHA-256

2.6.1 Pra-pemrosesan (*Preprocessing*)

1. Pengisian Pesan (*Message Padding*)

   Pesan yang telah dikodekan dalam bentuk bilangan biner, kemudian disisipkan angka 1 pada pesan awal dan menambahkan bit-bit penunjang yakni angka 0 hingga panjang pesan sepadan dengan 448 modulo 512. Panjang pesan asli tersebut selanjutnya ditambahkan sebagai angka biner 64 bit. Setelah itu didapatkan panjang pesan saat ini adalah kelipatan 512 bit. Tujuan dari *padding* ini adalah untuk memastikan bahwa pesan *padded* adalah kelipatan 512. *Padding* dapat dimasukkan sebelum komputasi hash dimulai pada pesan, atau pada waktu lain selama komputasi hash sebelum memproses blok yang akan berisi padding (Dang, 2015).

2. Penguraian (*Parsing*)

   Pesan yang telah dilakukan proses *padding* kemudian dibagi menjadi N buah blok 512 bit. Karena 512 bit blok input dapat dinyatakan dengan enam belas 32-bit, 32 bit pertama blok pesan i dapat dinyatakan dengan $M_0^{(i)}$, 32 bit selanjutnya adalah $M_1^{(i)}$, dan seterusnya hingga $M_{15}^{(i)}$.

3. Inisialisasi Nilai Hash ($H^{(0)}$)

Nilai dari inisialisasi nilai hash adalah 32 bit pertama dan didapatkan dari bagian pecahan akar kuadrat dari delapan bilangan prima pertama. Rumus tersebut dapat dituliskan sebagai berikut:

$$H^{(0)} = \left(\sqrt{n} - 1\right) \times 2^{32}. \quad (2.6)$$

Keterangan:

$H^{(0)}$ = Nilai hash

n = Bilangan prima

2.6.2 Komputasi Hash (*Hash Computation*)

1. Menyiapkan Jadwal Pesan (W_t)

Jadwal pesan untuk $0 \leq t \leq 15$ adalah:

$$W_t = M_t^i. \quad (2.7)$$

Jadwal pesan untuk $16 \leq t \leq 63$ adalah:

$$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}. \quad (2.8)$$

Keterangan:

W_t = Jadwal pesan

M_t^i = Blok pesan

2. Inisialisasi Delapan Variabel Kerja

Variabel kerja tersebut adalah a, b, c, d, e, f, g, dan h, dengan nilai hash ke-$(i - 1)$.

$$a = H_0^{(i-1)}.$$

$$b = H_1^{(i-1)}.$$

$$c = H_2^{(i-1)}.$$

$$d = H_3^{(i-1)}.$$

$$e = H_4^{(i-1)}.$$

$$f = H_5^{(i-1)}.$$

$$g = H_6^{(i-1)}.$$

$$h = H_7^{(i-1)}.$$

3. Konstanta SHA-256 (K_t)

Nilai konstanta dari SHA-256 didapatkan dari 32 bit pertama dari bagian fraksional dari akar kubik dari 64 bilangan prima pertama. Rumus tersebut dapat dituliskan sebagai berikut:

$$K_t = \left(\sqrt[3]{n} - 1\right) \times 2^{32}. \quad (2.9)$$

Keterangan:

K_t = Konstanta

n = Bilangan prima

Nilai n merupakan 64 bilangan prima pertama yang akan dikalkulasikan.

4. Fungsi SHA-256

$$Ch(e, f, g) = (e \wedge f) \oplus (\neg e \wedge g) \quad (2.10)$$

$$Maj(a, b, c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c) \quad (2.11)$$

$$\Sigma_0(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x) \quad (2.12)$$

$$\Sigma_1(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x) \quad (2.13)$$

$$\sigma_0(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x) \quad (2.14)$$

$$\sigma_1(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x) \quad (2.15)$$

Keterangan:

Ch = Fungsi *Choice*

Maj = Fungsi *Majority*

$ROTR$ = *Rotate right* (Perputaran ke kanan)

SHR = *Shift Right* (Pergeseran ke kanan)

5. Proses perhitungan pada $t = 0$ hingga $t = 63$ untuk melakukan update pada variabel kerja dengan memuat unsur jadwal pesan (W_t) dan nilai konstanta (K)

$$Temp_1 = h + \Sigma_1(e) + Ch(e, f, g) + K_t + W_t \quad (2.16)$$

$$Temp_2 = \Sigma_0(a) + Maj(a, b, c) \quad (2.17)$$

$$h = g \quad (2.18)$$

$$g = f \quad (2.19)$$

$$f = e \quad (2.20)$$

$$e = d + Temp_1 \quad (2.21)$$

$$d = c \quad (2.22)$$

$$c = b \quad (2.23)$$

$$b = a \quad (2.24)$$

$$a = Temp_1 + Temp_2 \quad (2.25)$$

6. Menghitung nilai hash antara nilai ke-*i* pada H^i

$$H_0^{(i)} = a + H_0^{(i)} \quad (2.26)$$

$$H_1^{(i)} = b + H_1^{(i)} \quad (2.27)$$

$$H_2^{(i)} = c + H_2^{(i)} \quad (2.28)$$

$$H_3^{(i)} = d + H_3^{(i)} \quad (2.29)$$

$$H_4^{(i)} = e + H_4^{(i)} \quad (2.30)$$

$$H_5^{(i)} = f + H_5^{(i)} \quad (2.31)$$

$$H_6^{(i)} = g + H_6^{(i)} \quad (2.32)$$

$$H_7^{(i)} = h + H_7^{(i)} \quad (2.33)$$

7. Hasil Keluaran SHA256 (*Output*)

Selanjutnya, keluaran SHA256 didapatkan dari penggabungan delapan variabel yang sebelumnya telah dilakukan komputasi.

$$H_0^N||H_1^N||H_2^N||H_3^N||H_4^N||H_5^N||H_6^N||H_7^N$$

2.7 Kajian Integrasi dengan Al-Qur'an

Ayat Al-Qur'an yang memuat terkait pembahasan pada penelitian ini adalah QS. Al-Mu'minun ayat 8-11 yang memiliki arti:

"Dan (sungguh beruntung) orang yang memelihara amanat-amanat (yang dipikulnya) dan janjinya, serta orang yang memelihara shalatnya. Mereka itulah orang yang akan mewarisi, (yakni) yang akan mewarisi (surga) Firdaus. Mereka kekal di dalamnya."

QS. Al-Mu'minun ayat 8 – 9 menyebutkan ciri-ciri orang beriman yang telah dijanjikan suatu kemenangan bagi manusia yang melaksanakannya. Ciri-ciri orang beriman yang termuat dalam QS. Al-Mu'minun ayat 8 yakni orang-orang

yang dapat menjaga amanat-amanat dan janjinya (Quthb, 2000). Amanat adalah sesuatu yang dipercayakan atau dititipkan kepada orang lain, pesan, nasihat yang baik dan berguna dari orang tua, perintah, dan wejangan. Amanat memilki dua macam, yaitu amanat umum dan amanat pribadi. Amanat umum adalah tugas yang diberikan oleh Allah atas dasar perikemanusiaan seluruhnya. Sehingga menjaga rahasia, ikhlas dalam memberikan anjuran kepada orang yang meminta pendapat dan menyampaikan pesan kepada pihak yang semestinya (sesuai dengan apa yang diminta oleh orang yang berpesan) juga termasuk amanah. Sedangkan amanat khusus merupakan tugas kita masing-masing berdasarkan kesanggupan diri, nasib, dan bakat. Menyadari bahwa dirinya hanya berkewajiban menjaga barang atau harta tersebut untuk tidak sampai rusak atau hilang dan tidak ada hak untuk menggunakan barang atau harta tersebut hingga dikembalikan kepada orang yang menitipkan (Abidin & Khairudin, 2017).

Amanat yang paling terdepan adalah amanat fitrah yang telah diciptakan oleh Allah kepada manusia untuk selalu lurus dan searah dengan Pencipta kehidupan yang merupakan sumber fitrah tersebut. Orang-orang yang beriman selalu menjaga amanat terbesar tersebut. Sehingga, mereka tidak pernah membiarkan fitrah mereka melenceng dari keistiqomahan mereka. Janji yang pertama juga adalah janji fitrah yang telah ditetapkan oleh Allah atas fitrah manusia dengan ketentuan iman kepada wujud-Nya dan tauhid-Nya. Setiap janji yang diikrarkan oleh setiap mukmin, pasti dia menjadikan Allah sebagai saksi di dalamnya (Quthb, 2000).

Ciri-ciri orang beriman selanjutnya yang dituliskan dalam QS. Al-Mu’minun ayat 9 yakni orang-orang yang memelihara sholatnya. Karakter orang

beriman sesungguhnya telah diawali dengan sholat dan diakhiri pula dengan sholat untuk menunjukkan tingginya martabat dan tingkatannya dalam membina iman. Karena, sholat adalah representasi ibadah yang paling sempurna dari ibadah-ibadah yang ditujukan kepada Allah (Quthb, 2000).

Allah memberi kehendak kepada orang-orang beriman yang berjalan di jalan yang ditentukan-Nya untuk sampai pada ujung target yang diperuntukkan bagi mereka karena gaya hidup duniawi tidak mungkin sempurna bagi manusia. Yaitu, surga firdaus, negeri yang abadi dan tidak akan pernah rusak, damai tanpa rasa takut sedikitpun serta ketenangan yang tanpa gangguan sedikitpun (Quthb, 2000).

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan dalam penelitian implementasi algoritma ElGamal dan fungsi *hash* SHA-256 pada data *electronic voting* (*e-voting*) adalah penelitian kualitatif.

3.2 Data dan Sumber Data

Data yang digunakan pada penelitian ini adalah data Nomor Induk Mahasiswa (NIM) Matematika angkatan 2018 Universitas Islam Negeri Maulana Malik Ibrahim Malang sebanyak 116 mahasiswa. Data NIM yang digunakan pada penelitian didapat berdasarkan informasi mahasiswa yang memiliki NIM paling tinggi pada angkatan 2018. Pada data NIM tersebut akan dilakukan percobaan pada NIM pertama dan untuk selanjutnya akan dituliskan pada lampiran penelitian. Pilihan yang diberikan pada proses *voting* adalah pilihan A dan pilihan B. Pada masing-masing NIM akan diberikan perlakuan apabila pengguna memilih pilihan A atau pengguna memilih pilihan B akan menghasilkan nilai *ciphertext* yang berbeda. Algoritma ElGamal yang membutuhkan nilai kunci privat akan dilakukan penggabungan angka ketiga dengan jumlah dari dua digit terakhir pada NIM pengguna. Sedangkan nilai acak bilangan bulat merupakan gabungan dari jumlah dua digit terakhir NIM pengguna dengan pilihan yang diberikan oleh pengguna dalam karakter ASCII. Apabila jumlah dua digit terakhir pada NIM terdiri dari dua digit, maka nilai tersebut akan dijumlahkan kembali hingga mendapatkan satu angka.

3.3 Tahapan Penelitian

Pada penelitian ini proses enkripsi dan dekripsi pilihan yang diberikan oleh pemilih menggunakan algoritma ElGamal. Sedangkan NIM masing-masing pengguna akan dilakukan perubahan ke dalam bentuk *message digest* menggunakan fungsi hash SHA-256. Penelitian dilakukan dengan tahapan sebagai berikut.

1. Proses hashing menggunakan algoritma SHA-256 dilakukan terlebih dahulu pada identitas pengguna yakni Nomor Induk Mahasiswa (NIM).
    a. Mengubah plainteks NIM kedalam bentuk bilangan biner, kemudian menyisipkan angka 1 pada pesan awal dan menambahkan angka 0 hingga panjang pesan sama dengan 448 modulo 512 sebagai bit penunjang.
    b. Panjang pesan asli selanjutnya ditambahkan sebagai angka biner 64 bit sehingga panjang pesan menjadi 512 bit.
    c. Pesan yang telah dilakukan proses padding kemudian dibagi menjadi enam belas blok dari 32 bit dan dari yang pertama dinyatakan dengan $M_0^{(i)}, M_1^{(i)}, M_2^{(i)}, \dots, M_{15}^{(i)}$.
    d. Menginisialisasi nilai hash dengan rumus (2.6) pada 8 bilangan prima pertama dan masing-masing diberikan label $H_0^{(0)}, H_1^{(0)}, H_2^{(0)}, \dots, H_7^{(0)}$.
    e. Menyiapkan jadwal pesan (W_t) sesuai dengan rumus (2.7) dan (2.8).
    f. Menginisialisasi 8 variabel kerja $a, b, c, d, e, f, g,$ dan h, dengan nilai hash ke $(i-1)$.
    g. Menyiapkan nilai konstanta SHA-256 dari 64 bilangan prima pertama dengan rumus (2.9) dan didapatkan nilai konstanta tersebut dengan label $K_0, K_1, K_2, \dots, K_{63}$.

h. Melakukan perhitungan fungsi hash $Ch(e,f,g)$, $Maj(a,b,c)$, $\Sigma_0(x)$, $\Sigma_1(x)$, $\sigma_0(x)$, dan $\sigma_1(x)$.

i. Melakukan proses perhitungan pada $t = 0$ hingga $t = 63$ untuk melakukan update pada variabel kerja dengan rumus (2.16) hingga (2.25).

j. Menghitung nilai hash antara nilai ke-*i* pada H^i dengan menjumlahkan variabel kerja pada perhitungan yang terakhir dengan inisialisasi nilai hash dan didapatkan nilai seperti pada rumus (2.26) hingga (2.33).

k. Nilai hash yang telah dijumlahkan diubah kedalam bentuk heksadesimal dan digabungkan untuk mendapatkan *message digest* dari plainteks awal.

l. Selanjutnya dilakukan autentikasi nilai hash yang telah didapatkan dengan nilai hash yang terdapat pada query data pemilih.

m. Nilai hash yang bersesuaian dapat melakukan proses selanjutnya yakni mengirimkan pilihannya.

2. Proses enkripsi menggunakan algoritma ElGamal digunakan pada enkripsi hasil voting yang dikirimkan oleh pengguna.

a. Menentukan bilangan prima p secara acak dengan rentang bilangan 1000-5000.

b. Menentukan nilai akar primitif terkecil (g) dari bilangan prima yang digunakan dengan syarat $g < p$.

c. Menentukan kunci privat (x) dengan syarat $1 \leq x \leq p - 2$. Nilai x diperoleh dari penggabungan angka ketiga dengan jumlah dua bilangan

terakhir dari NIM pengguna. Contoh pembentukan kunci privat adalah sebagai berikut:

i) NIM pengguna adalah 18610036

ii) Bilangan ke-3 adalah 6

iii) Bilangan ke-7 adalah 3

iv) Bilangan ke-8 adalah 6

v) Jumlah bilangan ke-7 dan ke-8 adalah 9

vi) Kunci privat dari NIM 18610036 adalah 69

d. Menghitung $y = g^x \, mod \, p$ untuk mendapatkan kunci publik. Kemudian akan didapatkan hasil kunci publik (y, g, p) dan kunci privat (x, p).

e. Menyusun plainteks menjadi blok-blok yakni m_1, m_2, dan seterusnya. Plainteks yang akan dienkripsi terdiri dari bilangan ketujuh pada NIM dilanjutkan dengan pilihan dari pengguna dan diakhir dengan bilangan kedelapan pada NIM. Contoh penyusunan plainteks adalah seperti berikut:

i) NIM pengguna adalah 18610036

ii) Bilangan ke-7 adalah 3

iii) Bilangan ke-8 adalah 6

iv) Pengguna mengirimkan pilihan A

v) Susunan plainteks adalah “3A6”

f. Menentukan nilai acak k dengan k terdapat dalam $1 \leq k \leq p - 2$. Nilai acak k didapatkan dari penggabungan jumlah dua bilangan terakhir

pada NIM dengan karakter pilihan dalam bentuk ASCII. Contoh penentuan nilai k sebagai berikut:

i) NIM pengguna adalah 18610036

ii) Jumlah dua bilangan terakhir pada NIM: 3+6=9

iii) Pengguna mengirimkan pilihan A. Bilangan ASCII untuk A adalah 65

iv) Susunan nilai k adalah: 965

g. Melakukan enkripsi pada plainteks yang telah disusun dengan rumus enkripsi elgamal pada (2.2) dan (2.3).

h. Nilai enkripsi pada setiap karakter dituliskan secara berurutan dalam bentuk: $a_1, b_1, a_2, b_2, a_3, b_3$.

i. Selanjutnya nilai tersebut dikirimkan pada query sesuai dengan pilihan yang dikirimkan pengguna.

3. Proses dekripsi pesan dilakukan pada ciphertext yang telah dikirimkan oleh pemilih dan telah disimpan dalam database.

a. Mengambil bilangan prima p secara acak yang telah digunakan pada proses enkripsi.

b. Mengambil bilangan acak x sebagai kunci privat.

c. Menghitung $c = a^{p-1-x} \; mod \; p$ dengan menggunakan kunci privat x dan bilangan prima p yang telah ditentukan pada saat enkripsi.

d. Menghitung plaintext m dengan persamaan $m = b.c \; mod \; p$.

e. Diperoleh plaintext seperti semula.

f. Selanjutnya dilakukan perhitungan hasil voting yang telah dikirimkan.

BAB IV

HASIL PENELITIAN

4.1 Proses Enkripsi dengan SHA-256

Proses enkripsi yang dilakukan pada identitas yang digunakan oleh pengguna yakni Nomor Induk Mahasiswa (NIM) dengan menggunakan fungsi hash SHA-256. Fungsi hash dengan SHA-256 akan dibagi menjadi dua proses yaitu yang pertama adalah pra-pemrosesan dan yang kedua adalah komputasi hash. Adapun langkah-langkah dari kedua proses tersebut adalah sebagai berikut.

4.1.1 Pra-pemrosesan (*Preprocessing*)

Proses pertama pada SHA-256 adalah pra-pemrosesan dengan 3 tahapan permulaan yaitu pengisian pesan, penguraian pesan, dan melakukan inisialisasi nilai hash. Penjelasan dari masing-masing tahapan adalah sebagai berikut.

1. Pengisian Pesan (*Message Padding*)

   Pesan inputan terlebih dahulu dikonversi kedalam bentuk bilangan biner. Selanjutnya dapat dilakukan pengisian pesan (*padding*). Daftar Nomor Induk Mahasiswa (NIM) yang digunakan dalam percobaan adalah:

Tabel 4.1 Daftar NIM Pemilih

No.	NIM	No.	NIM	No.	NIM
1	18610001	41	18610041	81	18610081
2	18610002	42	18610042	82	18610082
3	18610003	43	18610043	83	18610083
4	18610004	44	18610044	84	18610084
5	18610005	45	18610045	85	18610085

No.	NIM	No.	NIM	No.	NIM
6	18610006	46	18610046	86	18610086
7	18610007	47	18610047	87	18610087
8	18610008	48	18610048	88	18610088
9	18610009	49	18610049	89	18610089
10	18610010	50	18610050	90	18610090
11	18610011	51	18610051	91	18610091
12	18610012	52	18610052	92	18610092
13	18610013	53	18610053	93	18610093
14	18610014	54	18610054	94	18610094
15	18610015	55	18610055	95	18610095
16	18610016	56	18610056	96	18610096
17	18610017	57	18610057	97	18610097
18	18610018	58	18610058	98	18610098
19	18610019	59	18610059	99	18610099
20	18610020	60	18610060	100	18610100
21	18610021	61	18610061	101	18610101
22	18610022	62	18610062	102	18610102
23	18610023	63	18610063	103	18610103
24	18610024	64	18610064	104	18610104
25	18610025	65	18610065	105	18610105
26	18610026	66	18610066	106	18610106
27	18610027	67	18610067	107	18610107

No.	NIM	No.	NIM	No.	NIM
28	18610028	68	18610068	108	18610108
29	18610029	69	18610069	109	18610109
30	18610030	70	18610070	110	18610110
31	18610031	71	18610071	111	18610111
32	18610032	72	18610072	112	18610112
33	18610033	73	18610073	113	18610113
34	18610034	74	18610074	114	18610114
35	18610035	75	18610075	115	18610115
36	18610036	76	18610076	116	18610116
37	18610037	77	18610077		
38	18610038	78	18610078		
39	18610039	79	18610079		
40	18610040	80	18610080		

Pesan masukan tersebut selanjutnya dikodekan ke dalam bentuk bilangan biner yang terdapat pada ASCII dan didapatkan hasil sebagai berikut:

Tabel 4.2 Tabel NIM dalam Biner

No.	NIM	Karakter dalam Biner ASCII
1	18610001	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110001
2	18610002	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110010
3	18610003	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110011

No.	NIM	Karakter dalam Biner ASCII
4	18610004	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110100
5	18610005	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110101
6	18610006	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110110
7	18610007	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110111
8	18610008	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00111000
9	18610009	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00111001
10	18610010	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110000

Pada NIM ke-11 hingga NIM ke-116 dilakukan proses yang sama yaitu mengkodekan pesan masukan kedalam bentuk bilangan biner dan akan dituliskan pada lampiran 1. Selanjutnya pesan masukan yang telah dikodekan ke dalam bentuk bilangan biner disisipkan angka 1 pada awal pesan dan menambahkan angka 0 sebagai bit penunjang hingga sepadan dengan 448 modulo 512.

Percobaan pada NIM ke-1:

00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110001 10000000 00000000

00000000 00000000

Proses menyisipkan angka 1 dan menambahkan angka 0 dilakukan kepada semua pesan masukan, yaitu NIM ke-2 hingga NIM ke-116. Percobaan pada NIM pertama, selanjutnya panjang pesan asli tersebut ditambahkan sebagai angka biner untuk mendapatkan ukuran pesan kelipatan 512. Pesan asli yang diberikan memiliki delapan karakter. Apabila dituliskan dalam bilangan desimal akan dikalikan dengan delapan bit sehingga didapatkan 64 bit. Nilai desimal tersebut kemudian dibagi dengan dua untuk mendapatkan hasil dalam bentuk bilangan biner. Perhitungan dari bilangan desimal menjadi bilangan biner adalah sebagai berikut.

$\frac{64}{2}$ = 32 sisa 0 *Least Significant Bit* (LSB)

$\frac{32}{2}$ = 16 sisa 0

$\frac{16}{2}$ = 8 sisa 0

$\frac{8}{2}$ = 4 sisa 0

$\frac{4}{2}$ = 2 sisa 0

$\frac{2}{2}$ = 1 sisa 0

$\frac{1}{2}$ = 0 sisa 1 *Most Significant Bit* (MSB)

Sehingga bilangan desimal 64 apabila dikonversi ke dalam biner akan menjadi 1000000. Bilangan biner tersebut diletakkan paling akhir pada susunan pesan dalam biner. NIM pertama saat ini setelah ditambahkan bilangan biner 1000000 adalah sebagai berikut.

00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110001 10000000 01000000

Pesan yang telah dilakukan *padding*, selanjutnya akan dilakukan penguraian pesan.

2. Penguraian (*Parsing*)

   Pesan yang telah dituliskan dalam bentuk bilangan biner sepanjang 512 bit selanjutnya akan dibagi menjadi 16 buah oleh 32 bit. Pesan yang diuraikan pada 32 bit pertama akan diberikan label $M_0^{(i)}$, pesan pada 32 bit kedua akan diberikan label $M_1^{(i)}$, pesan pada 32 bit ketiga akan diberikan label $M_2^{(i)}$, dan seterusnya hingga pesan pada 32 bit keenam belas akan diberikan label $M_{15}^{(i)}$. Penguraian pesan yang dilakukan pada NIM pertama dapat dituliskan sebagai berikut.

$M_0^{(i)}$ = 00110001 00111000 00110110 00110001

$M_1^{(i)}$ = 00110000 00110000 00110000 00110001

$M_2^{(i)}$ = 10000000 00000000 00000000 00000000

$M_3^{(i)}$ = 00000000 00000000 00000000 00000000

$M_4^{(i)}$ = 00000000 00000000 00000000 00000000

$M_5^{(i)}$ = 00000000 00000000 00000000 00000000

$M_6^{(i)}$ = 00000000 00000000 00000000 00000000

$M_7^{(i)}$ = 00000000 00000000 00000000 00000000

$M_8^{(i)}$ = 00000000 00000000 00000000 00000000

$M_9^{(i)}$ = 00000000 00000000 00000000 00000000

$M_{10}^{(i)}$ = 00000000 00000000 00000000 00000000

$M_{11}^{(i)}$ = 00000000 00000000 00000000 00000000

$M_{12}^{(i)}$ = 00000000 00000000 00000000 00000000

$M_{13}^{(i)}$ = 00000000 00000000 00000000 00000000

$M_{14}^{(i)}$ = 00000000 00000000 00000000 00000000

$M_{15}^{(i)}$ = 00000000 00000000 00000000 01000000

Proses penguraian blok pesan dilakukan pada semua data NIM yang terdaftar dengan cara yang sama.

3. Inisialisasi Nilai Hash ($H^{(0)}$)

Proses selanjutnya pada tahap pertama adalah menentukan inisialisasi nilai hash ($H^{(0)}$) pada delapan bilangan prima pertama dengan rumus sebagai berikut.

$$H^{(0)} = \left(\sqrt{n} - 1\right) \times 2^{32}.$$

Karena bilangan prima pertama adalah 2, maka nilai hash dari bilangan prima tersebut adalah:

$$H_0^{(0)} = \left(\sqrt{2} - 1\right) \times 2^{32} = 1.41421356237330951.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $6a09e667$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

$$0110\ 1010\ 0000\ 1001\ 1110\ 0110\ 0110\ 0111.$$

Bilangan prima kedua adalah 3, maka nilai hash dari bilangan prima tersebut adalah:

$$H_1^{(0)} = \left(\sqrt{3} - 1\right) \times 2^{32} = 1.73205080756887772.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $bb67ae85$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

$$1011\ 1011\ 0110\ 0111\ 1010\ 1110\ 1000\ 0101.$$

Bilangan prima ketiga adalah 5, maka nilai hash dari bilangan prima tersebut adalah:

$$H_2^{(0)} = \left(\sqrt{5} - 1\right) \times 2^{32} = 2.23606797749979.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $3c6ef372$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

$$0011\ 1100\ 0110\ 1110\ 1111\ 0011\ 0111\ 0010.$$

Bilangan prima keempat adalah 7, maka nilai hash dari bilangan prima tersebut adalah:

$$H_3^{(0)} = \left(\sqrt{7} - 1\right) \times 2^{32} = 2.6457513110645907.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $a54ff53a$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

$$1010\ 0101\ 0100\ 1111\ 1111\ 0101\ 0011\ 1010.$$

Bilangan prima kelima adalah 11, maka nilai hash dari bilangan prima tersebut adalah:

$$H_4^{(0)} = \left(\sqrt{11} - 1\right) \times 2^{32} = 3.3166247903554.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $510e527f$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

$$0101\ 0001\ 0000\ 1110\ 0101\ 0010\ 0111\ 1111.$$

Bilangan prima keenam adalah 13, maka nilai hash dari bilangan prima tersebut adalah:

$$H_5^{(0)} = \left(\sqrt{13} - 1\right) \times 2^{32} = 3.605551275463989.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $9b05688c$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

1001 1011 0000 0101 0110 1000 1000 1100.

Bilangan prima ketujuh adalah 17, maka nilai hash dari bilangan prima tersebut adalah:

$$H_6^{(0)} = \left(\sqrt{17} - 1\right) \times 2^{32} = 4.123105625617661.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $1f83d9ab$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

0001 1111 1000 0011 1101 1001 1010 1011.

Bilangan prima kedelapan adalah 19, maka nilai hash dari bilangan prima tersebut adalah:

$$H_7^{(0)} = \left(\sqrt{19} - 1\right) \times 2^{32} = 4.358898943540674.$$

Nilai desimal tersebut selanjutnya diubah ke dalam bentuk bilangan heksadesimal menjadi $5be0cd19$. Nilai heksadesimal tersebut dapat dituliskan dalam bentuk bilangan biner menjadi:

0101 1011 1110 0000 1100 1101 0001 1001.

4.1.2 Komputasi Hash (*Hash Computation*)

Proses komputasi hash merupakan proses inti pada SHA-256 yang pada akhirnya akan menghasilkan keluaran (*output*) sepanjang 64 bit. Terdapat 7 tahapan dalam proses komputasi hash oleh SHA-256. Proses selanjutnya adalah sebagai berikut.

1. Menyiapkan Jadwal Pesan (W_t)

   Jadwal pesan (*Message Schedule*) dibuat sebanyak 64 baris di mana setiap baris berisi 32 bit. Jadwal pesan akan berhubungan dengan perhitungan pada fungsi hash SHA-256. Jadwal pesan baris ke-1 (W_0) merupakan blok pesan

yang telah diuraikan pada tahap pra-pemrosesan $M_0^{(i)}$. Jadwal pesan pada baris ke-2 (W_2) hingga ke-15 (W_{15}) merupakan blok pesan yang telah diuraikan pada tahap penguraian dan berlabel $M_1^{(i)}$ hingga $M_{15}^{(i)}$. Selanjutnya pada baris ke-17 (W_{16}) hingga ke-64 (W_{63}) merupakan susunan bilangan biner 0 sebanyak 32 bit pada setiap baris. Jadwal pesan dapat dituliskan sebagai berikut.

Tabel 4.3 Persiapan Jadwal Pesan

W_t	Pesan
W_0	00110001 00111000 00110110 00110001
W_1	00110000 00110000 00110000 00110001
W_2	10000000 00000000 00000000 00000000
W_3	00000000 00000000 00000000 00000000
W_4	00000000 00000000 00000000 00000000
W_5	00000000 00000000 00000000 00000000
W_6	00000000 00000000 00000000 00000000
W_7	00000000 00000000 00000000 00000000
W_8	00000000 00000000 00000000 00000000
W_9	00000000 00000000 00000000 00000000
W_{10}	00000000 00000000 00000000 00000000
W_{11}	00000000 00000000 00000000 00000000
W_{12}	00000000 00000000 00000000 00000000
W_{13}	00000000 00000000 00000000 00000000
W_{14}	00000000 00000000 00000000 00000000

W_t	**Pesan**
W_{15}	00000000 00000000 00000000 01000000
W_{16}	00000000 00000000 00000000 00000000
W_{17}	00000000 00000000 00000000 00000000
W_{18}	00000000 00000000 00000000 00000000
W_{19}	00000000 00000000 00000000 00000000
W_{20}	00000000 00000000 00000000 00000000
W_{21}	00000000 00000000 00000000 00000000
W_{22}	00000000 00000000 00000000 00000000
W_{23}	00000000 00000000 00000000 00000000
W_{24}	00000000 00000000 00000000 00000000
W_{25}	00000000 00000000 00000000 00000000
W_{26}	00000000 00000000 00000000 00000000
W_{27}	00000000 00000000 00000000 00000000
W_{28}	00000000 00000000 00000000 00000000
W_{29}	00000000 00000000 00000000 00000000
W_{30}	00000000 00000000 00000000 00000000
W_{31}	00000000 00000000 00000000 00000000
W_{32}	00000000 00000000 00000000 00000000
W_{33}	00000000 00000000 00000000 00000000
W_{34}	00000000 00000000 00000000 00000000
W_{35}	00000000 00000000 00000000 00000000
W_{36}	00000000 00000000 00000000 00000000

W_t	**Pesan**
W_{37}	00000000 00000000 00000000 00000000
W_{38}	00000000 00000000 00000000 00000000
W_{39}	00000000 00000000 00000000 00000000
W_{40}	00000000 00000000 00000000 00000000
W_{41}	00000000 00000000 00000000 00000000
W_{42}	00000000 00000000 00000000 00000000
W_{43}	00000000 00000000 00000000 00000000
W_{44}	00000000 00000000 00000000 00000000
W_{45}	00000000 00000000 00000000 00000000
W_{46}	00000000 00000000 00000000 00000000
W_{47}	00000000 00000000 00000000 00000000
W_{48}	00000000 00000000 00000000 00000000
W_{49}	00000000 00000000 00000000 00000000
W_{50}	00000000 00000000 00000000 00000000
W_{51}	00000000 00000000 00000000 00000000
W_{52}	00000000 00000000 00000000 00000000
W_{53}	00000000 00000000 00000000 00000000
W_{54}	00000000 00000000 00000000 00000000
W_{55}	00000000 00000000 00000000 00000000
W_{56}	00000000 00000000 00000000 00000000
W_{57}	00000000 00000000 00000000 00000000
W_{58}	00000000 00000000 00000000 00000000

W_t	Pesan
W_{59}	00000000 00000000 00000000 00000000
W_{60}	00000000 00000000 00000000 00000000
W_{61}	00000000 00000000 00000000 00000000
W_{62}	00000000 00000000 00000000 00000000
W_{63}	00000000 00000000 00000000 00000000

Jadwal pesan selanjutnya pada urutan ke-17 (W_{16}) hingga ke-64 (W_{63}) didapatkan dengan kalkulasi sebagai berikut.

$$W_t = W_{(t-7)} + W_{(t-16)} + \sigma_0(W_{(t-15)}) + \sigma_1(W_{(t-2)}).$$

Di mana:

$$\sigma_0(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x)$$

$$\sigma_1(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$$

Sebelum mendapatkan nilai W_{16}, akan dilakukan kalkulasi terlebih dahulu pada σ_0 dan σ_1. Nilai σ_0 adalah sebagai berikut.

$$W_{16-15} = W_1 = 00110001\ 00111000\ 00110110\ 00110001$$

Tabel 4.4 Operasi XOR pada W_1

Identitas	Nilai	Operasi	Ket.
$ROTR^7$ (W_1)	01100010 01100000 01100000 01100000		
$ROTR^{18}$ (W_1)	00001100 00001100 01001100 00001100	$\oplus$	XOR
SHR^3 (W_1)	00000110 00000110 00000110 00000110	$\oplus$	XOR
σ_0	01101000 01101010 00101010 01101010		Hasil

Nilai σ_1 adalah sebagai berikut.

$$W_{16-2} = W_{14} = 00000000\ 00000000\ 00000000\ 00000000$$

Tabel 4.5 Operasi XOR pada W_{14}

Identitas	**Nilai**	**Operasi**	**Ket.**
$ROTR^{17}$ (W_{14})	00000000 00000000 00000000 00000000		
$ROTR^{19}$ (W_{14})	00000000 00000000 00000000 00000000	$\oplus$	XOR
SHR^{10} (W_{14})	00000000 00000000 00000000 00000000	$\oplus$	XOR
σ_1	00000000 00000000 00000000 00000000		Hasil

Setelah didapatkan σ_0 dan σ_1, maka nilai dari W_{16} adalah sebagai berikut.

Tabel 4.6 Jumlah pada W_{16}

Identitas	**Nilai**	**Operasi**	**Ket.**
W_9	00000000 00000000 00000000 00000000		
W_0	00110001 00111000 00110110 00110001	+	Tambah
σ_0	01101000 01101010 00101010 01101010	+	Tambah
σ_1	00000000 00000000 00000000 00000000	+	Tambah
W_{16}	10011001 10100010 01100000 10011011		Hasil

Kemudian dilanjutkan pada nilai W_{17} dengan terlebih dahulu mengkalkulasikan σ_0 dan σ_1 sebagai berikut.

$$W_{17-15} = W_2 = 10000000\ 00000000\ 00000000\ 00000000$$

Tabel 4.7 Operasi XOR pada W_2

Identitas	Nilai	Operasi	Ket.
$ROTR^7$ (W_2)	00000001 00000000 00000000 00000000		
$ROTR^{18}$ (W_2)	00000000 00000000 00100000 00000000	$\oplus$	XOR
SHR^3 (W_2)	00010000 00000000 00000000 00000000	$\oplus$	XOR
σ_0	00010001 00000000 00100000 00000000		Hasil

$$W_{17-2} = W_{15} = 00000000\ 00000000\ 00000000\ 01000000$$

Tabel 4.8 Operasi XOR pada W_{15}

Identitas	Nilai	Operasi	Ket.
$ROTR^{17}$ (W_{15})	00000000 00100000 00000000 00000000		
$ROTR^{19}$ (W_{15})	00000000 00001000 00000000 00000000	$\oplus$	XOR
SHR^{10} (W_{15})	00000000 00000000 00000000 00000000	$\oplus$	XOR
σ_1	00000000 00101000 00000000 00000000		Hasil

Setelah didapatkan σ_0 dan σ_1, maka nilai dari W_{17} adalah sebagai berikut.

Tabel 4.9 Jumlah pada W_{17}

Identitas	Nilai	Operasi	Ket.
W_{10}	00000000 00000000 00000000 00000000		
W_1	00110000 00110000 00110000 00110001	+	Tambah
σ_0	00010001 00000000 00100000 00000000	+	Tambah

σ_1	00000000 00101000 00000000 00000000	+	Tambah
W_{17}	01000001 01011000 01010000 00110001		Hasil

Nilai pada W_{18} hingga W_{63} didapatkan dengan cara kalkulasi serupa secara berulang. Berdasarkan kalkulasi yang telah dilakukan, didapatkan nilai W_{18} hingga W_{63} pada pesan masukan "18610001" yang telah diperbarui adalah sebagai berikut.

Tabel 4.10 Hasil Jadwal Pesan

W_t	Pesan
W_0	00110001 00111000 00110110 00110001
W_1	00110000 00110000 00110000 00110001
W_2	10000000 00000000 00000000 00000000
W_3	00000000 00000000 00000000 00000000
W_4	00000000 00000000 00000000 00000000
W_5	00000000 00000000 00000000 00000000
W_6	00000000 00000000 00000000 00000000
W_7	00000000 00000000 00000000 00000000
W_8	00000000 00000000 00000000 00000000
W_9	00000000 00000000 00000000 00000000
W_{10}	00000000 00000000 00000000 00000000
W_{11}	00000000 00000000 00000000 00000000

W_t	**Pesan**
W_{12}	00000000 00000000 00000000 00000000
W_{13}	00000000 00000000 00000000 00000000
W_{14}	00000000 00000000 00000000 00000000
W_{15}	00000000 00000000 00000000 01000000
W_{16}	10011001 10100010 01100000 10011011
W_{17}	01000001 01011000 01010000 00110001
W_{18}	11111100 01111000 11010111 01111101
W_{19}	00100010 00001110 11011110 10010011
W_{20}	01110001 01101110 01011111 10000110
W_{21}	10110100 10010011 01110110 11110001
W_{22}	11100100 00101111 10101101 01001101
W_{23}	01101111 00101110 00101000 10100001
W_{24}	01100100 10001110 10010101 10101010
W_{25}	11001101 11011000 00101001 01110101
W_{26}	10111010 10001000 00111100 00000110
W_{27}	10000011 00010101 10001000 11100011
W_{28}	11001110 01000000 10011111 00001011
W_{29}	01011001 01111101 00100001 11010111

W_t	Pesan
W_{30}	01001011 10010101 00110111 01111000
W_{31}	01010101 01110111 10011101 01111101
W_{32}	00100011 01100010 00111110 11110101
W_{33}	00001010 10001101 01111110 10001000
W_{34}	01101101 11011100 01011100 10011110
W_{35}	10010110 00010101 10101111 11111010
W_{36}	10011000 11111010 00110010 10110110
W_{37}	11001110 01101111 00011111 00000100
W_{38}	01011101 11100110 10101111 01000110
W_{39}	11111100 00010100 00100011 10010000
W_{40}	11101001 11110110 01110111 00100000
W_{41}	11100110 01011111 10100011 11101101
W_{42}	11111011 01001000 01111011 00100000
W_{43}	01101011 01011000 11110100 11110100
W_{44}	10111101 01100010 11110111 00001000
W_{45}	11010000 10011101 11011110 01111011
W_{46}	10000100 01001110 10110110 01110000
W_{47}	11110101 11001100 10010101 01000010

W_t	Pesan
W_{48}	11100110 01111111 11011011 01110100
W_{49}	00000100 01010001 11111001 01111111
W_{50}	01111100 00110010 11111100 10111101
W_{51}	00001001 10001101 00110001 11101011
W_{52}	01100010 11111110 00111100 10010110
W_{53}	10111110 01011110 11101011 11111110
W_{54}	01100100 00010011 00011010 01100111
W_{55}	01001100 01101000 00110110 11011001
W_{56}	00001100 10101010 11111101 10011001
W_{57}	11000000 10001110 11010111 10011110
W_{58}	11111110 11010010 11110000 00101011
W_{59}	00111010 01111000 10000001 11000100
W_{60}	00111100 00011101 10001011 01101001
W_{61}	11100001 10100010 01100111 11100110
W_{62}	00000011 10010000 01100011 00101001
W_{63}	10000101 10001101 10010100 10011110

Nilai jadwal pesan pada blok pesan NIM kedua hingga NIM ke-116 memiliki proses yang sama pada saat menentukan hasil dari jadwal pesan.

2. Inisialisasi Delapan Variabel Kerja

   Inisialisasi variabel kerja pada mulanya berasal dari inisialisasi nilai hash ke-$(i-1)$ yang dapat dituliskan dalam bentuk bilangan biner. Pada variabel kerja terdapat perhitungan yang dilakukan berulang sesuai dengan rumus fungsi inisialisasi variabel kerja dan selanjutnya variabel kerja akan selalu diperbarui. Variabel kerja yang pertama dalam bentuk bilangan biner dan heksadesimal adalah sebagai berikut.

   $a = H_0^{(i-1)} = 01101010\ 00001001\ 11100110\ 01100111 = 6a09e667$

   $b = H_1^{(i-1)} = 10111011\ 01100111\ 10101110\ 10000101 = bb67ae85$

   $c = H_2^{(i-1)} = 00111100\ 01101110\ 11110011\ 01110010 = 3c6ef372$

   $d = H_3^{(i-1)} = 10100101\ 01001111\ 11110101\ 00111010 = a54ff53a$

   $e = H_4^{(i-1)} = 01010001\ 00001110\ 01010010\ 01111111 = 510e527f$

   $f = H_5^{(i-1)} = 10011011\ 00000101\ 01101000\ 10001100 = 9b05688c$

   $g = H_6^{(i-1)} = 00011111\ 10000011\ 11011001\ 10101011 = 1f83d9ab$

   $h = H_7^{(i-1)} = 01011011\ 11100000\ 11001101\ 00011001 = 5be0cd19$

3. Konstanta SHA256

   Nilai konstanta pada SHA-256 menggambarkan 32-bit pertama bagian pecahan akar pangkat tiga dari 64 bilangan prima pertama. Nilai konstanta SHA-256 dapat dituliskan dalam bentuk bilangan biner maupun heksadesimal. Berikut merupakan tahapan dalam menentukan konstanta dari SHA-256 yang dituliskan dalam bentuk bilangan biner maupun heksadesimal. Rumus yang digunakan dalam menentukan konstanta SHA-256 adalah sebagai berikut.

$$K_t = \left(\sqrt[3]{n} - 1\right) \times 2^{32}.$$

Karena bilangan prima pertama adalah 2, maka nilai konstanta dari bilangan tersebut adalah:

$$K_0 = \left(\sqrt[3]{2} - 1\right) \times 2^{32} = 1116352408.8404646.$$

Selanjutnya, nilai tersebut diubah ke dalam bentuk bilangan heksadesimal menjadi $428a2f98$.

Bilangan prima kedua adalah 3, maka nilai konstanta dari bilangan tersebut adalah:

$$K_1 = \left(\sqrt[3]{3} - 1\right) \times 2^{32} = 1899447441.1403713$$

Selanjutnya, nilai tersebut diubah ke dalam bentuk bilangan heksadesimal menjadi 71374491.

Nilai konstanta selanjutnya didapatkan dari bilangan prima ketiga hingga bilangan prima ke-64 dengan cara yang sama. Nilai konstanta pada SHA-256 secara keseluruhan adalah sebagai berikut.

Tabel 4.11 Nilai Konstanta SHA-256

Konstanta	Nilai	Konstanta	Nilai
K_0	$428a2f98$	K_{32}	$27b70a85$
K_1	71374491	K_{33}	$2e1b2138$
K_2	$b5c0fbcf$	K_{34}	$4d2c6dfc$
K_3	$e9b5dba5$	K_{35}	$53380d13$
K_4	$3956c25b$	K_{36}	$650a7354$
K_5	$59f111f1$	K_{37}	$766a0abb$
K_6	$923f82a4$	K_{38}	$81c2c92e$
K_7	$ab1c5ed5$	K_{39}	$92722c85$

Konstanta	Nilai	Konstanta	Nilai
K_8	$d807aa98$	K_{40}	$a2bfe8a1$
K_9	$12835b01$	K_{41}	$a81a664b$
K_{10}	$243185be$	K_{42}	$c24b8b70$
K_{11}	$550c7dc3$	K_{43}	$c76c51a3$
K_{12}	$72be5d74$	K_{44}	$d192e819$
K_{13}	$80deb1fe$	K_{45}	$d6990624$
K_{14}	$9bdc06a7$	K_{46}	$f40e3585$
K_{15}	$c19bf174$	K_{47}	$106aa070$
K_{16}	$e49b69c1$	K_{48}	$19a4c116$
K_{17}	$efbe4786$	K_{49}	$1e376c08$
K_{18}	$0fc19dc6$	K_{50}	$2748774c$
K_{19}	$240ca1cc$	K_{51}	$34b0bcb5$
K_{20}	$2de92c6f$	K_{52}	$391c0cb3$
K_{21}	$4a7484aa$	K_{53}	$4ed8aa4a$
K_{22}	$5cb0a9dc$	K_{54}	$5b9cca4f$
K_{23}	$76f988da$	K_{55}	$682e6ff3$
K_{24}	$983e5152$	K_{56}	$748f82ee$
K_{25}	$a831c66d$	K_{57}	$78a5636f$
K_{26}	$b00327c8$	K_{58}	$84c87814$
K_{27}	$bf597fc7$	K_{59}	$8cc70208$
K_{28}	$c6e00bf3$	K_{60}	$90beffa$
K_{29}	$d5a79147$	K_{61}	$a4506ceb$

Konstanta	Nilai	Konstanta	Nilai
K_{30}	$06ca6351$	K_{62}	$bef9a3f7$
K_{31}	14292967	K_{63}	$c67178f2$

4. Kalkulasi nilai $t = 0$ hingga $t = 63$ dengan Rumus Fungsi Hash SHA-256

   Rumus fungsi SHA-256 akan digunakan dalam melakukan pembaruan variabel kerja. Pada proses ini akan melibatkan barisan jadwal pesan (W_t) dan nilai konstanta (K_t). Sebelum variabel kerja diperbarui, variabel kerja pada mulanya memiliki nilai serupa dengan nilai hash SHA-256 seperti yang telah dituliskan pada tahap inisialisasi delapan variabel kerja. Proses ini akan dilakukan pada NIM pertama yakni "18610001" sebagai penjelas pada tahap perhitungan menggunakan fungsi hash SHA-256. Pembaruan variabel kerja adalah sebagai berikut.

$$h = g.$$

$$g = f.$$

$$f = e.$$

$$e = d + Temp1.$$

$$d = c.$$

$$c = b.$$

$$b = a.$$

$$a = Temp_1 + Temp_2.$$

   Pembaruan variabel kerja pertama pada a merupakan jumlah dari *temporary word* pertama ($Temp1$) dan *temporary word* kedua ($Temp2$). Dimana $Temp1$ adalah sebagai berikut:

$$Temp_1 = h + \Sigma_1(e) + Ch(e,f,g) + K_t + W_t.$$

Sehingga terlebih dahulu akan dilakukan kalkulasi pada $\Sigma_1(e)$ dan $Ch(e,f,g)$.

$$Temp_1 = h + \Sigma_1(e) + Ch(e,f,g) + K_t + W_t.$$

$$\Sigma_1(e) = ROTR^6(e) \oplus ROTR^{11}(e) \oplus ROTR^{25}(e).$$

Di mana nilai variabel kerja e adalah:

$$e = 01010001\ 00001110\ 01010010\ 01111111.$$

Tabel 4.12 Operasi XOR pada Σ_1

Identitas	Nilai	Operasi	Ket.
$ROTR^6(e)$	11111101 01000100 00111001 01001001		
$ROTR^{11}(e)$	01001111 11101010 00100001 11001010	$\oplus$	XOR
$ROTR^{25}(e)$	10000111 00101001 00111111 10101000	$\oplus$	XOR
$\Sigma_1^{\{256\}}$	00110101 10000111 00100111 00101011		Hasil

$$Ch(e,f,g) = (e \wedge f) \oplus (\neg e \wedge g).$$

Tabel 4.13 Operasi XOR pada fungsi Choice

Identitas	Nilai	Operasi	Ket.
$(e \wedge f)$	00010001 00000100 01000000 00001100		
$(\neg e \wedge g)$	00001110 10000001 10001001 10000000	$\oplus$	XOR
$Ch(e,f,g)$	00011111 10000101 11001001 10001100		Hasil

$$Temp_1 = h + \Sigma_1(e) + Ch(e,f,g) + K_t + W_t.$$

Tabel 4.14 Operasi penjumlahan pada $Temp_1$

Identitas	Nilai	Ope-rasi	Ket.
h	01011011 11100000 11001101 00011001		
$\Sigma_1^{\{256\}}(e)$	00110101 10000111 00100111 00101011	+	Tambah
$Ch(e,f,g)$	00011111 10000101 11001001 10001100	+	Tambah
$K_0^{\{256\}}$	01000010 10001010 00101111 10011000	+	Tambah
W_0	00110001 00111000 00110110 00110001	+	Tambah
$Temp_1$	00100100 10110000 00100011 10011001		Hasil

$$Temp_2 = \Sigma_0(a) + Maj(a,b,c).$$

$$\Sigma_0(a) = ROTR^2(a) \oplus ROTR^{13}(a) \oplus ROTR^{22}(a).$$

Di mana nilai variabel a adalah:

$$a = 01101010\ 00001001\ 11100110\ 01100111.$$

Tabel 4.15 Operasi XOR pada Σ_0

Identitas	Nilai	Operasi	Ket.
$ROTR^2(a)$	11011010 10000010 01111001 10011001		
$ROTR^{13}(a)$	00110011 00111011 01010000 01001111	$\oplus$	XOR
$ROTR^{22}(a)$	00100111 10011001 10011101 10101000	$\oplus$	XOR
Σ_0	11001110 00100000 10110100 01111110		Hasil

$$Maj(a,b,c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c).$$

Tabel 4.16 Operasi XOR pada $Majority$

Identitas	Nilai	Operasi	Ket.
$(a \wedge b)$	00101010 00000001 10100110 00000101		
$(a \wedge c)$	00101000 00001000 11100010 01100010	$\oplus$	XOR
$(b \wedge c)$	00111000 01100110 10100000 00000000	$\oplus$	XOR
$Maj(a,b,c)$	00111010 01101111 11100110 01100111		Hasil

Tabel 4.17 Operasi penjumlahan pada $Temp_2$

Identitas	Nilai	Ope-rasi	Ket.
$\Sigma_0^{\{256\}}$	11001110 00100000 10110100 01111110		
$Maj(a,b,c)$	00111010 01101111 11100110 01100111	$+$	Tambah
$Temp_2$	00001000 10010000 10011010 11100101		Hasil

Pembaruan variabel kerja pada a adalah sebagai berikut.

$$a = Temp_1 + Temp_2.$$

Tabel 4.18 Operasi penjumlahan pada variabel a

Identitas	Nilai	Operasi	Ket.
$Temp_1$	00100100 10110000 00100011 10011001		
$Temp_2$	00001000 10010000 10011010 11100101	$+$	Tambah
a	00101101 01000000 10111110 01111110		Nilai

Pembaruan variabel kerja pada e adalah sebagai berikut.

$$e = d + Temp_1.$$

Tabel 4.19 Operasi penjumlahan pada variabel *e*

Identitas	Nilai	Operasi	Ket.
d	10100101 01001111 11110101 00111010		
$Temp_1$	00100100 10110000 00100011 10011001	+	Tambah
e	11001010 00000000 00011000 11010011		Hasil

Variabel kerja secara keseluruhan pada perhitungan pertama adalah sebagai berikut.

$$a = 00101101\ 01000000\ 10111110\ 01111110$$

$$b = 01101010\ 00001001\ 11100110\ 01100111$$

$$c = 10111011\ 01100111\ 10101110\ 10000101$$

$$d = 00111100\ 01101110\ 11110011\ 01110010$$

$$e = 11001010\ 00000000\ 00011000\ 11010011$$

$$f = 01010001\ 00001110\ 01010010\ 01111111$$

$$g = 10011011\ 00000101\ 01101000\ 10001100$$

$$h = 00011111\ 10000011\ 11011001\ 10101011$$

Selanjutnya, delapan variabel kerja tersebut akan dilakukan pembaruan dengan menggunakan nilai W_1 dan K_1. Sebelum melakukan pembaruan pada delapan variabel kerja untuk yang kedua kali, dibutuhkan nilai $Temp_1$ dan $Temp_2$ dengan rumus sebagai berikut.

$$Temp_1 = h + \Sigma_1(e) + Ch(e, f, g) + \mathrm{K}_1 + W_1.$$

Nilai pada $Temp_1$ memuat nilai $\Sigma_1(e)$ dan $Ch(e,f,g)$, sehingga akan dilakukan terlebih dahulu pada unsur tersebut dengan cara sebagai berikut.

$$\Sigma_1(e) = ROTR^6(e) \oplus ROTR^{11}(e) \oplus ROTR^{25}(e).$$

Di mana nilai variabel e adalah:

$$e = 11001010\ 00000000\ 00011000\ 11010011.$$

Tabel 4.20 Operasi XOR pada Σ_1

Identitas	Nilai	Operasi	Ket.
$ROTR^6(e)$	01001111 00101000 00000000 01100011		
$ROTR^{11}(e)$	00011010 01111001 01000000 00000011	$\oplus$	XOR
$ROTR^{25}(e)$	00000000 00001100 01101001 11100101	$\oplus$	XOR
Σ_1	01010101 01011101 00101001 10000101		Hasil

$$Ch(e,f,g) = (e \wedge f) \oplus (\neg e \wedge g).$$

Tabel 4.21 Operasi XOR pada fungsi *Choice*

Identitas	Nilai	Operasi	Ket.
$(e \wedge f)$	01000000 00000000 00010000 01010011		
$(\neg e \wedge g)$	00010001 00000101 01100000 00001100	$\oplus$	XOR
$Ch(e,f,g)$	01010001 00000101 01110000 01011111		Hasil

Sehingga didapatkan nilai pada $Temp_1$ sebagai berikut.

Tabel 4.22 Operasi penjumlahan pada $Temp_1$

Identitas	Nilai	Operasi	Ket.
h	00011111 10000011 11011001 10101011		
$\Sigma_1(e)$	01010101 01011101 00101001 10000101	+	Tambah

Identitas	Nilai	Operasi	Ket.
$Ch(e,f,g)$	01010001 00000101 01110000 01011111	+	Tambah
K_1	01110001 00110111 01000100 10010001	+	Tambah
W_1	00110000 00110000 00110000 00110001	+	Tambah
$Temp_1$	01100111 01001101 11101000 01010001		Hasil

Selanjutnya pada nilai $Temp_2$ dengan cara sebagai berikut akan dilalkukan kalkulasi terlebih dahulu pada nilai $\Sigma_0(a)$ dan $Maj(a,b,c)$.

$$Temp_2 = \Sigma_0(a) + Maj(a,b,c)$$

$$\Sigma_0(a) = ROTR^2(a) \oplus ROTR^{13}(a) \oplus ROTR^{22}(a)$$

$$a = 00101101\ 01000000\ 10111110\ 01111110$$

Tabel 4.23 Operasi XOR pada Σ_0

Identitas	Nilai	Operasi	Ket.
$ROTR^2(a)$	10001011 01010000 00101111 10011111		
$ROTR^{13}(a)$	11110011 11110001 01101010 00000101	$\oplus$	XOR
$ROTR^{22}(a)$	00000010 11111001 11111000 10110101	$\oplus$	XOR
Σ_0	01111010 01011000 10111101 00101111		Hasil

$$Maj(a,b,c) = (a \wedge b) \oplus (a \wedge c) \oplus (b \wedge c).$$

Tabel 4.24 Operasi XOR pada fungsi *Majority*

Identitas	Nilai	Operasi	Ket.
$(a \wedge b)$	00101000 00000000 10100110 01100110		
$(a \wedge c)$	00101001 01000000 10101110 00000100	$\oplus$	XOR

Identitas	Nilai	Ope-rasi	Ket.
$(b \wedge c)$	00101010 00000001 10100110 00000101	$\oplus$	XOR
$Maj(a, b, c)$	00101011 01000001 10101110 01100111		Hasil

Tabel 4.25 Operasi penjumlahan pada $Temp_2$

Identitas	Nilai	Ope-rasi	Ket.
Σ_0	01111010 01011000 10111101 00101111		
$Maj(a, b, c)$	00101011 01000001 10101110 01100111	$+$	Tambah
$Temp_2$	10100101 10011010 01101011 10010110		Hasil

$Temp_1$ dan $Temp_2$ akan dijumlahkan untuk melakukan pembaruan pada variabel kerja a.

$$a = Temp_1 + Temp_2.$$

Tabel 4.26 Operasi penjumlahan pada variabel a

Identitas	Nilai	Operasi	Ket.
$Temp_1$	01100111 01001101 11101000 01010001		
$Temp_2$	10100101 10011010 01101011 10010110	$+$	Tambah
a	00001100 11101000 01010011 11100111		Hasil

Pembaruan pada variabel kerja e didapatkan dari penjumlahan variabel kerja d dengan $Temp_1$ dan mendapatkan hasil sebagai berikut.

$$e = d + Temp_1.$$

Tabel 4.27 Operasi penjumlahan pada variabel *e*

Identitas	Nilai	Operasi	Ket.
d	00111100 01101110 11110011 01110010		
$Temp_1$	01100111 01001101 11101000 01010001	$+$	Tambah
e	10100011 10111100 11011011 11000011		Hasil

Delapan variabel kerja selanjutnya setelah dilakukan pembaruan kedua adalah sebagai berikut.

$$a = 00001100111010000101001111100111$$

$$b = 00101101010000001011111001111110$$

$$c = 01101010000010011110011001100111$$

$$d = 10111011011001111010111010000101$$

$$e = 10100011101111001101101111000011$$

$$f = 11001010000000000001100011010011$$

$$g = 01010001000011100101001001111111$$

$$h = 10011011000001010110100010001100$$

Pembaruan delapan variabel kerja tersebut akan terus dilakukan dengan memuat nilai W_2 hingga W_{63} dan nilai K_2 hingga K_{63}. Hasil akhir pada berbaruan delapan variabel kerja adalah sebagai berikut.

$$a = 11111011010110001100011000011000$$

$$b = 00100101101101100111000001101111$$

$$c = 10000010100000011011100001101000$$

$$d = 00011100011110000000100101101010$$

$$e = 00101001000000011000001011011110$$

$$f = 11000100111101110001011101011101$$

$$g = 10100101001100011101011011010011$$

$$h = 00111000101010101111001010111111$$

5. Hasil Keluaran SHA-256 (*Output*)

   Hasil keluaran SHA-256 didapat dari penjumlahan nilai hash mula-mula dengan delapan variabel kerja pada pembaruan terakhir. Fungsi penjumlahan tersebut adalah sebagai berikut.

$$H_0^{(i)} = a + H_0^{(i-1)}.$$

$$H_1^{(i)} = b + H_1^{(i-1)}.$$

$$H_2^{(i)} = c + H_2^{(i-1)}.$$

$$H_3^{(i)} = d + H_3^{(i-1)}.$$

$$H_4^{(i)} = e + H_4^{(i-1)}.$$

$$H_5^{(i)} = f + H_5^{(i-1)}.$$

$$H_6^{(i)} = g + H_6^{(i-1)}.$$

$$H_7^{(i)} = h + H_7^{(i-1)}.$$

   Keluaran pada SHA-256 selanjutnya dituliskan secara berurutan untuk mendapatkan nilai hash dari SHA-256 dengan urutan sebagai berikut.

$$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}.$$

   Berdasarkan perhitungan pada variabel kerja yang dilakukan secara berulang dan telah diketahui nilai hash mula-mula dalam biner, maka keluaran dari SHA-256 pada pesan “18610001” adalah sebagai berikut.

$$H_0^{(1)} = a + H_0^{(1-1)}.$$

Tabel 4.28 Operasi Penjumlahan variabel a dengan H_0

Identitas	Nilai	Operasi	Ket.
a	10111101 00010110 00011000 11110000		
$H_0^{(1-1)}$	01101010 00001001 11100110 01100111	+	Tambah
$H_0^{(1)}$	00100111 00011111 11111111 01010111		Hasil

$$H_1^{(2)} = b + H_1^{(2-1)}.$$

Tabel 4.29 Operasi Penjumlahan variabel b dengan H_1

Identitas	Nilai	Operasi	Ket.
b	11111011 01011000 11000110 00011000		
$H_1^{(2-1)}$	10111011 01100111 10101110 10000101	+	Tambah
$H_1^{(1)}$	10110110 11000000 01110100 10011101		Hasil

$$H_2^{(3)} = c + H_2^{(3-1)}.$$

Tabel 4.30 Operasi penjumlahan variabel c dengan H_2

Identitas	Nilai	Operasi	Ket.
c	00100101 10110110 01110000 01101111		
$H_2^{(3-1)}$	00111100 01101110 11110011 01110010	+	Tambah
$H_2^{(1)}$	01100010 00100101 01100011 11100001		Hasil

$$H_3^{(4)} = d + H_3^{(4-1)}.$$

Tabel 4.31 Operasi penjumlahan variabel d dengan H_3

Identitas	Nilai	Operasi	Ket.
d	10000010 10000001 10111000 01101000		
$H_3^{(4-1)}$	10100101 01001111 11110101 00111010	+	Tambah
$H_3^{(1)}$	00100111 11010001 10101101 10100010		Hasil

$$H_4^{(5)} = e + H_4^{(5-1)}.$$

Tabel 4.32 Operasi penjumlahan variabel e dengan H_4

Identitas	Nilai	Operasi	Ket.
e	11001000 11110011 10101001 01000101		
$H_4^{(5-1)}$	01010001 00001110 01010010 01111111	+	Tambah
$H_4^{(1)}$	00011010 00000001 11111011 11000100		Hasil

$$H_5^{(6)} = f + H_5^{(6-1)}.$$

Tabel 4.33 Operasi penjumlahan variabel f dengan H_5

Identitas	Nilai	Operasi	Ket.
f	00101001 00000001 10000010 11011110		
$H_5^{(6-1)}$	10011011 00000101 01101000 10001100	+	Tambah
$H_5^{(5)}$	11000100 00000110 11101011 01101010		Hasil

$$H_6^{(7)} = g + H_6^{(7-1)}.$$

Tabel 4.34 Operasi penjumlahan variabel g dengan H_6

Identitas	Nilai	Operasi	Ket.
g	11000100 11110111 00010111 01011101		
$H_6^{(7-1)}$	00011111 10000011 11011001 10101011	+	Tambah
$H_6^{(6)}$	11100100 01111010 11110001 00001000		Hasil

$$H_7^{(8)} = h + H_7^{(8-1)}.$$

Tabel 4.35 Operasi penjumlahan variabel h dengan H_7

Identitas	Nilai	Operasi	Ket.
h	10100101 00110001 11010110 11010011		
$H_7^{(8-1)}$	01011011 11100000 11001101 00011001	+	Tambah
$H_7^{(7)}$	00000001 00010010 10100011 11101100		Hasil

Hasil pada masing-masing nilai hash selanjutnya dikonversi kedalam bentuk bilangan heksadesimal dan mendapatkan nilai sebagai berikut.

$$H_0^{(1)} = 00100111\ 00011111\ 11111111\ 01010111 = 271fff57$$

$$H_1^{(1)} = 10110110\ 11000000\ 01110100\ 10011101 = b6c0749d$$

$$H_2^{(1)} = 01100010\ 00100101\ 01100011\ 11100001 = 622563e1$$

$$H_3^{(1)} = 00100111\ 11010001\ 10101101\ 10100010 = 27d1ada2$$

$$H_4^{(1)} = 00011010\ 00000001\ 11111011\ 11000100 = 1a01fbc4$$

$$H_5^{(5)} = 11000100\ 00000110\ 11101011\ 01101010 = c406eb6a$$

$$H_6^{(6)} = 11100100\ 01111010\ 11110001\ 00001000 = e471f108$$

$$H_7^{(7)} = 00000001\ 00010010\ 10100011\ 11101100 = 0112a3ec$$

Sehingga apabila disusun menjadi keluaran SHA-256 adalah sebagai berikut.

$$271fff57b6c0749d622563e127d1ada2$$

$$1a01fbc4c406eb6ae471f1080112a3ec.$$

Pada pesan masukan NIM “18610002” hingga NIM “18610116” dilakukan operasi perhitungan yang sama berawal dari tahap pra-pemrosesan hingga tahap komputasi fungsi hash. Hasil akhir pada NIM selanjutnya akan dituliskan pada lampiran 2.

4.2 Algoritma ElGamal

Proses enkripsi selanjutnya dilakukan pada pilihan yang dikirimkan oleh pengguna atau pemilih dengan menggunakan algoritma ElGamal. Algoritma ElGamal juga akan digunakan dalam proses dekripsi *ciphertext* pada hasil voting yang tersimpan pada database guna melakukan perhitungan hasil akhir pemilihan. Pada enkripsi dan dekripsi algoritma ElGamal akan menggunakan bilangan prima ribuan dan akar primitif terkecil dari bilangan prima yang digunakan. Sebelum melakukan proses enkripsi, algoritma elgamal memerlukan sebuah kunci publik dan kunci privat. Sehingga pada algoritma elgamal terdapat proses pembangkitan kunci.

4.2.1. Pembangkitan Kunci

Proses pada pembangkitan kunci akan menggunakan bilangan prima (p) ribuan serta beberapa kombinasi yang telah ditentukan pada bilangan acak yang nantinya akan digunakan. Bilangan prima pada proses pembangkitan kunci merupakan hasil pemilihan bilangan prima secara acak oleh bahasa pemrograman

python. Setiap pesan atau masukan yang akan di enkripsi memiliki bilangan prima yang berbeda. Setelah mendapatkan bilangan prima, akan dilakukan pemilihan nilai akar primitif (g) terkecil dari bilangan prima yang digunakan. Selain bilangan prima dan akar primitif, pembangkitan kunci juga memerlukan suatu bilangan yang digunakan sebagai kunci privat pengguna. Kunci privat (x) pada pembangkitan kunci merupakan kombinasi dari angka ketiga digabungkan dengan jumlah dua angka terakhir pada NIM. Proses pembangkitan kunci pada algoritma elgamal adalah sebagai berikut.

Pembangkitan kunci akan dilakukan pada NIM pertama pengguna yang terdaftar pada database yaitu "18610001". Bilangan prima yang digunakan adalah $p = 1553$. Akar primitif terkecil pada bilangan prima tersebut adalah $g = 3$. Kunci privat merupakan kombinasi angka ketiga pada NIM digabungkan dengan jumlah dua angka terakhir pada NIM, sehingga kunci privat yang digunakan adalah $x = 61$. Selanjutnya melakukan perhitungan menggunakan rumus dibawah ini untuk menemukan nilai dari kunci publik yang akan digunakan pada proses enkripsi.

$$y = g^x \ mod \ p$$

$$y = 3^{61} \ mod \ 1553 = 695.$$

Sehingga pasangan kunci publik untuk NIM "18610001" adalah (y, g, p) = (695,3,1551) dan pasangan kunci privat adalah (x, p) = $(61, 1553)$.

Pembangkitan kunci pada NIM kedua "18610002" dengan menggunakan bilangan prima $p = 2473$ dan akar primitif bilangan prima tersebut adalah $g = 5$. Kombinasi kunci privat yang digunakan adalah $x = 62$, sehingga pasangan kunci publik adalah sebagai berikut.

$$y = 5^{62} \ mod \ 2473 = 1922.$$

Maka, pasangan kunci publik pada NIM kedua adalah (y, g, p) = $(1922,5,2473)$ dan pasangan kunci privat adalah (x, p) = $(62,2473)$.

Pembangkitan kunci pada NIM selanjutnya dilakukan dengan cara yang sama untuk menghasilkan pasangan kunci publik dan kunci privat. Hasil pembangkitan kunci pada sepuluh NIM pertama adalah sebagai berikut.

Tabel 4.36 Nilai pembangkitan kunci publik ElGamal

NO.	NIM	p	g	x	k	(y, g, p)
1.	18610001	1553	3	61	165	(695,3,1553)
2.	18610002	2473	5	62	266	(1922,5,2473)
3.	18610003	4219	2	63	365	(162,2,4219)
4.	18610004	1399	13	64	466	(1280,13,1399)
5.	18610005	3191	11	65	565	(2127,11,3191)
6.	18610006	2741	2	66	666	(1188,2,2741)
7.	18610007	3041	3	67	765	(2137,3,3041)
8.	18610008	3001	14	68	866	(4,14,3001)
9.	18610009	2851	2	69	965	(2367,2,2851)
10.	18610010	3089	3	61	166	(594,3,3089)

Proses perhitungan pembangkitan kunci diterapkan pada seluruh NIM pengguna yang terdaftar dan akan dituliskan pada lampiran 3. Setelah melakukan pembangkitan kunci, didapatkan pasangan kunci publik dan kunci privat yang akan digunakan pada proses enkripsi dan dekripsi. Proses enkripsi dan dekripsi algoritma elgamal akan dijelaskan pada sub-bab selanjutnya.

4.2.2. Proses Enkripsi Algoritma ElGamal

Proses enkripsi menggunakan algoritma elgamal akan dilakukan pada hasil voting (pilihan) yang dikirimkan oleh pengguna atau pemilih. Pesan masukan yang akan di enkripsi merupakan gabungan dari penggunaan dua digit terakhir pada NIM pemilih dan disisipkan pilihan yang dikirimkan diantara dua digit tersebut. Pesan masukan tersebut dapat dituliskan sebagai berikut. NIM = 18610036, dua digit terakhir pada NIM tersebut adalah 3 dan 6. Apabila pemilih mengirimkan pilihan A, maka pesan masukan yang akan di enkripsi adalah "3A6". Apabila pemilih mengirimkan pilihan B, maka pesan masukan yang akan di enkripsi adalah "3B6".

Enkripsi menggunakan algoritma elgamal dalam prosesnya akan menggunakan bilangan prima, akar primitif, kunci publik, dan bilangan bulat acak. Enkripsi pada NIM pertama "18610001" menggunakan bilangan prima yang sama dengan bilangan prima pada saat melakukan pembangkitan kunci yaitu $p = 1553$ dan $g = 3$. Bilangan bulat acak merupakan gabungan dari jumlah dua digit NIM pemilih dengan pilihannya dalam ASCII. Proses enkripsi akan dilakukan pada masing-masing NIM dengan kemungkinan mengirimkan pilihan A dan B. Apabila NIM pertama memilih A, maka pesan masukannya adalah "0A1" dan nilai bilangan bulat acak yang terbentuk adalah $k = 165$. Apabila NIM pertama memilih B, maka pesan masukannya adalah "0B1" dan nilai bilangan bulat acak yang terbentuk adalah $k = 166$. Proses perhitungan *cipherteks* pada pilihan NIM pertama apabila memilih A adalah sebagai berikut.

$$a = g^k mod\ p = 3^{165}\ mod\ 1553 = 1235.$$

Pesan masukan yang terbentuk pada NIM pertama adalah penggunaan dua digit terakhir NIM dan disisipkan pilihan yang dikirimkan oleh pengguna. Pesan

masukan pada NIM pertama apabila memilih A adalah "$0A1$", maka hasil enkripsi pesan tersebut adalah sebagai berikut.

$$m_1 = \text{"0"} = 48 \rightarrow b = my^k \ mod \ p = (48)(695^{165}) \ mod \ 1553 = 968.$$

$$m_2 = \text{"A"} = 65 \rightarrow b = my^k \ mod \ p = (65)(695^{165}) \ mod \ 1553 = 1052.$$

$$m_3 = \text{"1"} = 49 \rightarrow b = my^k \ mod \ p = (49)(695^{165}) \ mod \ 1553 = 1247.$$

Maka, pasangan nilai enkripsi tersebut adalah:

$$a_1, b_1, a_2, b_2, a_3, b_3 = 1235{,}968{,}1235{,}1052{,}1235{,}1247.$$

Proses perhitungan *cipherteks* pada pilihan NIM pertama apabila memilih B adalah sebagai berikut.

$$a = g^k mod \ p = 3^{166} \ mod \ 1553 = 599.$$

Pesan masukan pada NIM pertama apabila memilih B adalah "$0B1$", maka hasil enkripsi pesan tersebut adalah sebagai berikut.

$$m_1 = \text{"0"} = 48 \rightarrow b = my^k \ mod \ p = (48)(695^{166}) \ mod \ 1553 = 311.$$

$$m_2 = \text{"B"} = 66 \rightarrow b = my^k \ mod \ p = (66)(695^{166}) \ mod \ 1553 = 1010.$$

$$m_3 = \text{"1"} = 49 \rightarrow b = my^k \ mod \ p = (49)(695^{166}) \ mod \ 1553 = 91.$$

Maka, pasangan nilai enkripsi tersebut adalah:

$$a_1, b_1, a_2, b_2, a_3, b_3 = 599{,}311{,}599{,}1010{,}599{,}91.$$

Enkripsi selanjutnya pada NIM pemilih kedua yakni "18610002" dengan bilangan prima $p = 2473$, akar primitif $g = 5$, bilangan bulat acak $k = 265$ apabila memilih A, kunci publik $y = 1922$, dan kunci privat $(x, p) = (62{,}2473)$. Hasil enkripsi pada pesan masukan $m = \text{"0}A2\text{"}$ adalah sebagai berikut.

$$a = g^k mod \ p = 5^{265} \ mod \ 2473 = 2245.$$

$$m_1 = \text{"0"} = 48 \rightarrow b = my^k \ mod \ p = (48)(1922^{265}) \ mod \ 2473 = 107.$$

$$m_2 = \text{"A"} = 65 \rightarrow b = my^k \ mod \ p = (65)(1992^{265}) \ mod \ 2473 = 1639.$$

$$m_3 = \text{"2"} = 50 \rightarrow b = my^k \ mod \ p = (50)(1992^{265}) \ mod \ 2473 = 1451.$$

Maka, pasangan nilai enkripsi pada NIM kedua adalah sebagai berikut.

$$a_1, b_1, a_2, b_2, a_3, b_3 = 2245{,}107{,}2245{,}1639{,}2245{,}1451.$$

Hasil enkripsi pada pesan masukan NIM kedua apabila memilih B dengan $m =$ "$0B2$" adalah sebagai berikut.

$$a = g^k mod \ p = 5^{266} \ mod \ 2473 = 1333.$$

$$m_1 = \text{"0"} = 48 \rightarrow b = my^k \ mod \ p = (48)(1922^{266}) \ mod \ 2473 = 395.$$

$$m_2 = \text{"B"} = 66 \rightarrow b = my^k \ mod \ p = (66)(1992^{266}) \ mod \ 2473 = 234.$$

$$m_3 = \text{"2"} = 50 \rightarrow b = my^k \ mod \ p = (50)(1992^{266}) \ mod \ 2473 = 1751.$$

Maka, pasangan nilai enkripsi pada NIM kedua adalah sebagai berikut.

$$a_1, b_1, a_2, b_2, a_3, b_3 = 1333{,}395{,}1333{,}234{,}1333{,}1751.$$

Enkripsi dilakukan pada NIM selanjutnya hingga akhir menggunakan perhitungan dan proses yang sama. Setelah didapatkan hasil dari enkripsi dan berupa *ciphertext*, maka proses selanjutnya adalah dekripsi atau pengembalian ke pesan semula. Berikut adalah hasil dari enkripsi pada pesan masukan hasil pemilihan yang diberikan oleh sepuluh pemilih pertama.

Tabel 4.37 Hasil enkripsi algoritma ElGamal

NO.	**NIM**	***Pesan***	a_1, b_1	a_2, b_2	a_3, b_3
1.	18610001	0A1	(1235,968)	(1235,1052)	(1235,1247)
		0B1	(599,311)	(599,1010)	(599,91)
2.	18610002	0A2	(2245,107)	(2245,1639)	(2245,1451)
		0B2	(1333,395)	(1333,234)	(1333,1751)
3.	18610003	0A3	(84,4200)	(84,3578)	(84,2353)

NO.	NIM	*Pesan*	a_1, b_1	a_2, b_2	a_3, b_3
		0B3	(168, 1141)	(168,3151)	(168,1476)
4.	18610036	0A4	(508,1157)	(508,1013)	(508,1370)
		0B4	(1008,818)	(1008,775)	(1008,653)
5.	18610005	0A5	(2032,939)	(2032,1471)	(2032,2034)
		0B5	(15,2878)	(15,1564)	(15,2513)
6.	18610006	0A6	(2354,389)	(2354,2297)	(2354,95)
		0B6	(1967,1644)	(1967,890)	(1967,479)
7.	18610007	0A7	(2581,2796)	(2581,3026)	(2581,923)
		0B7	(1661,2528)	(1661,435)	(1661,1883)
8	18610008	0A8	(67,932)	(67,1012)	(67,87)
		0B8	(938,727)	(938,2125)	(928,1805)
9	18610009	0A9	(1198,1903)	(1198,1840)	(1198,122)
		0B9	(2396,2672)	(2396,823)	(2396,322)
10	18610010	1A0	(141,2954)	(141,3036)	(141,246)
		1B0	(423,124)	(423,1680)	(423941)

Proses enkripsi pilihan yang diberikan oleh pemilih dilakukan dengan perhitungan yang sama. Pada percobaan NIM ke-11 hingga ke-116 dilakukan cara perhitungan yang sama dan akan dituliskan pada bagian lampiran 4.

4.2.3. Proses Dekripsi Algoritma ElGamal

Proses dekripsi merupakan proses pengembalian pasangan *ciphertext* (a, b) ke dalam bentuk pesan aslinya atau *plaintext*. Proses dekripsi memerlukan nilai sepasang kunci pribadi (x, p) dan nilai bilangan prima yang digunakan pada setiap karakter untuk mendekripsi *ciphertext*. Proses dekripsi pada *ciphertext* tersebut adalah sebagai berikut.

Nilai *ciphertext* pada NIM pertama "18610001" apabila memilih A adalah $(a_1, b_1) = (1235,968)$. Proses pengembalian pesan tersebut menggunakan rumus $m = b.c \ mod \ p$. Bilangan prima $p = 1553$ dan kunci privat $x = 61$, maka nilai c adalah sebagai berikut.

$$c_1 = a^{p-1-x} \ mod \ p = 1235^{1553-1-61} \ mod \ 1553 = 1091.$$

Selanjutnya nilai c dapat disubstitusikan pada persamaan berikut untuk menghitung nilai ASCII pada *ciphertext*.

$$m_1 = b_1.c_1 \ mod \ p = (968)(1091) mod \ 1553 = 48 = 0.$$

Ciphertext selanjutnya adalah $(a_2, b_2) = (1235,1052)$, maka nilai m_2 adalah sebagai berikut.

$$m_2 = b_2.c_1 \ mod \ p = (1052)(1091) mod \ 1553 = 65 = A.$$

Ciphertext ketiga adalah $(a_3, b_3) = (1235,1247)$, maka nilai m_3 adalah sebagai berikut

$$m_3 = b_3.c_1 \ mod \ p = (1247)(1091) mod \ 1553 = 49 = 1.$$

Nilai ketiga *ciphertext* setelah dilakukan dekripsi telah sesuai dengan pesan asli sebelum dilakukan enkripsi yaitu $m =$ "0A1".

Chipertext pertama pada NIM pertama apabila memilih B adalah $(a_1, b_1) = (599,311)$. Bilangan prima $p = 1553$ dan kunci privat $x = 61$, maka nilai *c* adalah sebagai berikut.

$$c_1 = a^{p-1-x}\ mod\ p = 599^{1553-1-61}\ mod\ 1553 = 120.$$

Selanjutnya nilai *c* dapat disubstitusikan pada persamaan berikut untuk menghitung nilai ASCII pada *ciphertext*.

$$m_1 = b_1.c_1\ mod\ p = (311)(120) mod\ 1553 = 48 = 0.$$

Ciphertext selanjutnya adalah $(a_2, b_2) = (599,1010)$, maka nilai m_2 adalah sebagai berikut.

$$m_2 = b_2.c_1\ mod\ p = (1010)(120) mod\ 1553 = 66 = B.$$

Ciphertext ketiga adalah $(a_3, b_3) = (599,91)$, maka nilai m_3 adalah sebagai berikut.

$$m_3 = b_3.c_1\ mod\ p = (91)(120) mod\ 1553 = 49 = 1.$$

Nilai ketiga *ciphertext* setelah dilakukan dekripsi telah sesuai dengan pesan asli sebelum dilakukan enkripsi yaitu $m = "0B1"$.

Pada NIM kedua apabila memilih A didapatkan *ciphertext* pertama adalah $(a_1, b_1) = (2245,107)$. Bilangan prima $p = 2473$ dan kunci privat $x = 62$, maka nilai c_2 adalah sebagai berikut.

$$c_2 = a^{p-1-x}\ mod\ p = 2245^{2473-1-62}\ mod\ 2473 = 2381.$$

Nilai *c* yang telah diperoleh disubstitusikan pada rumus $m = b.c\ mod\ p$. Sehingga nilai m_1 adalah sebagai berikut.

$$m_1 = b_1.c_1\ mod\ p = (107)(2381) mod\ 2473 = 48 = 0.$$

Ciphertext selanjutnya adalah $(a_2, b_2) = (2245,1639)$, maka nilai m_2 adalah sebagai berikut.

$$m_2 = b_2.c_1 \; mod \; p = (1639)(2381) mod \; 2473 = 65 = A.$$

Ciphertext ketiga adalah $(a_3, b_3) = (2245{,}1451)$, maka nilai m_3 adalah sebagai berikut

$$m_3 = b_3.c_1 \; mod \; p = (1451)(2381) mod \; 2473 = 50 = 2.$$

Nilai ketiga ciphertext setelah dilakukan dekripsi telah sesuai dengan pesan asli sebelum dilakukan enkripsi yaitu $m = \text{"0A2"}$.

Ciphertext selanjutnya dari pesan kedua dengan NIM "18610002" adalah $(a_1, b_1) = (1333{,}395)$. Bilangan prima $p = 2473$ dan kunci privat $x = 62$, maka nilai c_2 adalah sebagai berikut.

$$c_2 = a^{p-1-x} \; mod \; p = 1333^{2473-1-62} \; mod \; 2473 = 1966.$$

Nilai c yang telah diperoleh disubstitusikan pada rumus $m = b.c \; mod \; p$. Sehingga nilai m_1 adalah sebagai berikut.

$$m_1 = b_1.c_1 \; mod \; p = (395)(1966) mod \; 2473 = 48 = 0.$$

Ciphertext selanjutnya adalah $(a_2, b_2) = (1333{,}234)$, maka nilai m_2 adalah sebagai berikut.

$$m_2 = b_2.c_1 \; mod \; p = (234)(1966) mod \; 2473 = 66 = B.$$

Ciphertext ketiga adalah $(a_3, b_3) = (1333{,}1751)$, maka nilai m_3 adalah sebagai berikut

$$m_3 = b_3.c_1 \; mod \; p = (1751)(1966) mod \; 2473 = 50 = 2.$$

Nilai ketiga ciphertext setelah dilakukan dekripsi telah sesuai dengan pesan asli sebelum dilakukan enkripsi yaitu $m = \text{"0B2"}$.

Proses dekripsi pada setiap *ciphertext* dilakukan dengan cara yang sama hingga mendapatkan hasil yang sesuai dengan pesan masukan. Hasil pada proses

dekripsi pada sepuluh NIM pertama yang dilakukan menggunakan algoritma elgamal adalah sebagai berikut.

Tabel 4.38 Hasil dekripsi algoritma ElGamal

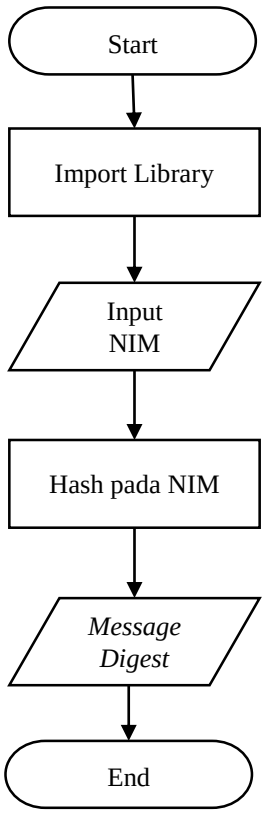
No.	NIM	(a_1, b_1)	m_1	(a_2, b_2)	m_2	(a_3, b_3)	m_3
1.	18610001	(1235,968)	0	(1235,1052)	A	(1235,1247)	1
		(599,311)	0	(599,1010)	B	(599,91)	1
2.	18610002	(2245,107)	0	(2245,1639)	A	(2245,1451)	2
		(1333,395)	0	(1333,234)	B	(1333,1751)	2
3.	18610003	(84,4200)	0	(84,3578)	A	(84,2353)	3
		(168, 1141)	0	(168,3151)	B	(168,1476)	3
4.	18610036	(508,1157)	0	(508,1013)	A	(508,1370)	4
		(1008,818)	0	(1008,775)	B	(1008,653)	4
5.	18610005	(2032,939)	0	(2032,1471)	A	(2032,2034)	5
		(15,2878)	0	(15,1564)	B	(15,2513)	5
6.	18610006	(2354,389)	0	(2354,2297)	A	(2354,95)	6
		(1967,1644)	0	(1967,890)	B	(1967,479)	6
7.	18610007	(2581,2796)	0	(2581,3026)	A	(2581,923)	7
		(1661,2528)	0	(1661,435)	B	(1661,1883)	7
8	18610008	(67,932)	0	(67,1012)	A	(67,87)	8
		(938,727)	0	(938,2125)	B	(928,1805)	8
9	18610009	(1198,1903)	0	(1198,1840)	A	(1198,122)	9
		(2396,2672)	0	(2396,823)	B	(2396,322)	9

No.	NIM	(a_1, b_1)	m_1	(a_2, b_2)	m_2	(a_3, b_3)	m_3
10	18610010	(141,2954)	1	(141,3036)	A	(141,246)	0
		(423,124)	1	(423,1680)	B	(423941)	0

Proses dekripsi dilakukan pada seluruh *ciphertext* yang telah dikirimkan oleh pengguna. Hasil dari proses dekripsi secara keseluruhan akan dituliskan pada bagian lampiran 5.

4.3 Flowchart Program Python

Pada penelitian ini digunakan bantuan suatu program dalam bahasa python guna memudahkan proses perhitungan pada setiap tahapnya. Alur program yang digunakan pada proses perhitungan fungsi hash SHA-256 adalah sebagai berikut.



Gambaran program yang dijalankan pada proses enkripsi dengan fungsi hash SHA-256.

```
# import semua library yang digunakan
import numpy as np
import pandas as pd
import random
from functools import wraps
from math import sqrt
import hashlib
```

Gambar 4.1 Import Library Pada Python

```
# Fungsi untuk melakukan hash pada semua data NIM dan menulisnya di file excel
for i in range(len(df)):
    nim = str(df['nim'][i])
    sha = hashlib.sha256(nim.encode(encoding='ascii'))
    df['sha'][i]=sha.hexdigest()
    df.to_excel('arange.xlsx')
```

Gambar 4.2 Proses Pada Fungsi Hash SHA-256

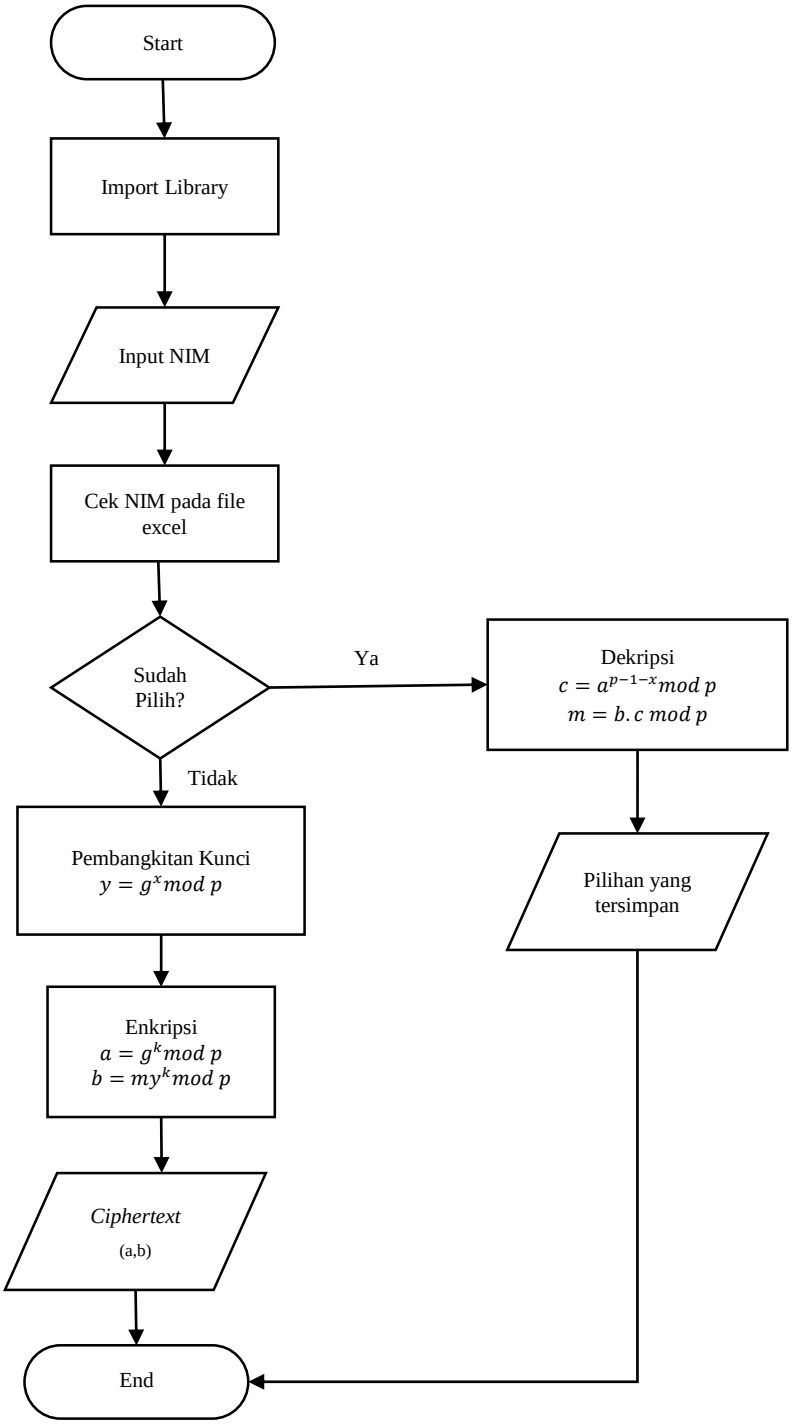
```
df= pd.read_excel('arange.xlsx', index_col=0)
df
```

	nim	sha	pilihan	publickey	privatekey	prima	g	k	a	b	dekripsi
0	18610001	271fff57b6c0749d622563e127d1ada21a01fbc4c406eb...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	18610002	0996d7bd977fb60fd9f1fcdff1add029b57db20719c7a2...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	18610003	32fc038613cf7a582f6b39dbe8d6001eb806671d61f006...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	18610004	6fa1640570cbd1f6ba0036453adec924d857c9738daf7c...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	18610005	8991bd4ea8fb3887fd23216b8e0dca04bd90e6d03d3052...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
111	18610112	77e42bedd0f4c4e1c551903df1063325740826cb28c50c...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
112	18610113	2d82a611cb62b5600882aba361429cf221b8e7bcf9a600...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
113	18610114	7100d1cfb4c71dfdd0fa5163e03fe929c7aa603dc734db...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
114	18610115	3f88ec64718c1d1d83bccf69def24001b43563941287ab...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
115	18610116	4c9d26366551e2e1d702fd9f7b64263b780ea93c459805...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

116 rows × 11 columns

Gambar 4.3 Hasil Fungsi Hash SHA-256 Pada NIM

Alur program selanjutnya pada perhitungan algoritma ElGamal melakukan pengecekan terhadap NIM yang terdaftar pada file excel. Apabila NIM yang dimasukkan belum mengirimkan pilihannya, program akan melanjutkan pada proses pemilihan. Apabila NIM yang dimasukkan telah mengirimkan pilihannya, maka program akan menampilkan pilihan yang telah dikirimkan pengguna.



Gambaran program yang dijalankan pada proses enkripsi hasil pemilihan yang dikirimkan oleh pemilih.

```
# Memasukkan NIM yang akan dicek
nim = str(input("Masukkan NIM: "))

# Lakukan hash dengan SHA
sha = hashlib.sha256(nim.encode(encoding='ascii'))


# Cari data pada file yang sesuai dengan NIM yang sudah di-hash
pil = df.loc[df['sha'] == sha.hexdigest()]
baris = pil.index.tolist()[0]

if pil['pilihan'].isnull().values.any():
    prime_list = primesInRange(100,200)
    randomPrime = random.choice(prime_list)
    p = int(randomPrime)
    df.at[baris,'prima'] = p
    g = int(findPrimitive(p))
    df.at[baris,'g'] = g
    x= random.randint(10,p-2)
    df.at[baris,'privatekey'] = x
    y = (int(g**x) % p)
    y = int(y)
    df.at[baris,'publickey'] = y
    enkripsi()
    print("Terima Kasih, Anda Sudah Memilih !")

else:
    dekrip = dekripsi()
    print("Mohon Maaf, Anda Sudah Memilih !")
```

```
Masukkan NIM: 18610004
Masukkan pilihan Anda : B
Terima Kasih, Anda Sudah Memilih !
```

Gambar 4.4 Proses Pemilihan dan Enkripsi dengan Algoritma ElGamal

4.4 Kajian Keislaman dengan Hasil Penelitian

Berdasarkan hasil yang diperoleh dari penelitian ini, pesan masukan yang pada mulanya dapat dibaca oleh seluruh pihak menjadi pesan yang tidak dapat terbaca secara langsung. Pengamanan pesan menjadi pesan yang tidak dapat terbaca secara langsung guna menyimpan identitas penting tiap pengguna atau tiap individu. Pesan yang telah diamankan hanya dapat diolah dan diketahui oleh pihak tertentu. Pihak tersebut memiliki tanggung jawab untuk menyimpan amanat dari orang lain. Dalam Al-Qur'an telah disebutkan keuntungan seseorang menjaga amanat yang diberikan kepadanya. Hal ini tertuang pada QS. Al-Mu'minun ayat 8-11 yang memiliki arti (Kemenag, 2019):

"Dan (sungguh beruntung) orang yang memelihara amanat-amanat (yang dipikulnya) dan janjinya, serta orang yang memelihara shalatnya. Mereka itulah orang yang akan mewarisi, (yakni) yang akan mewarisi (surga) Firdaus. Mereka kekal di dalamnya."

Pada ayat ini Allah telah menjelaskan keuntungan yang diperoleh manusia yang dapat menjaga setiap amanat yang diberikan kepadanya. Salah satu contoh amanat yang dijelaskan menurut Tafsir *Wajiz* adalah perkawinan, maka setiap orang harus memeliharanya dengan baik. Tidak hanya amanat dalam perkawinan, melainkan semua amanat. Allah juga memberikan keuntungan kepada manusia yang dapat menjaga sholatnya, diantaranya dengan memelihara waktu sholat pada awal waktu, serta memelihara pula rukun, wajib , dan sunahnya. Keuntungan yang diberikan Allah kepada manusia yang dapat menjaga amanat yang dipikulnya dan memelihara sholatnya adalah Surga Firdaus. Mereka akan kekal di dalam kenikmatan dan kebahagiaan tanpa gangguan sedikitpun (Kemenag, 2019).

Penjelasan tersebut apabila dikaitkan dengan menjaga setiap amanat yang kita dapatkan, hendaklah kita senantiasa menjaga amanat tersebut dengan baik serta tidak melakukan penyalahgunaan pada amanat yang telah diberikan. Dengan kita dapat menjaga amanat tersebut, tidak hanya kepercayaan orang lain terhadap kita yang kita terima, melainkan keuntungan untuk dapat menikmati Surga Firdaus dengan kekal pada masanya nanti.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian tersebut dapat ditarik kesimpulan bahwa data pemilihan umum dapat diamankan melalui algoritma ElGamal dan fungsi hash SHA-256. Fungsi hash SHA-256 yang memiliki tahap pra-pemrosesan dan komputasi hash dapat melakukan pengamanan pada identitas yang digunakan oleh pemilih yaitu Nomor Induk Mahasiswa (NIM). Hasil dari SHA-256 adalah sebuah karakter *message digest* yang memiliki panjang 64 bit, sehingga data yang telah diproses dengan fungsi hash SHA-256 tidak dapat dibaca oleh orang yang tidak berwenang. Enkripsi selanjutnya dilakukan pada hasil pemilihan yang dikirimkan oleh pemilih menggunakan enkripsi elgamal. Pada enkripsi elgamal terdiri dari tiga proses yaitu pembangkitan kunci, enkripsi, dan dekripsi. Pada tahap pembangkitan kunci dengan rumus $y = g^x \ mod \ p$ menghasilkan pasangan kunci publik yang digunakan pada tahap enkripsi. Enkripsi elgamal dilakukan dengan rumus $a = g^k \ mod \ p$ dan $b = m \ y^k \ mod \ p$. Hasil enkripsi memiliki bentuk ($a_1, b_1, a_2. b_2, ...$). Selanjutnya hasil enkripsi tersebut dilakukan proses dekripsi dengan rumus $m = b. c \ mod \ p$, dimana $c = a^{p-1-x} \ mod \ p$ untuk mendapatkan pesan aslinya.

5.2 Saran untuk Penelitian Selanjutnya

Pada penelitian ini hanya melakukan percobaan rumus fungsi hash SHA-256 dan algoritma ElGamal dalam melakukan enkripsi dan dekripsi pesan. Terdapat banyak kekurangan dalam penerapannya pada program pemilihan berbasis elektronik yang dapat digunakan oleh masyarakat. Sehingga diharapkan pada

penelitian selanjutnya, peneliti dapat melakukan implementasi fungsi hash serta enkripsi dan dekripsi menggunakan jenis fungsi hash dan algoritma lainnya dalam mengamankan pesan atau data *electronic voting*. Serta dapat menerapkannya pada suatu program pemilihan berbasis elektronik yang dapat digunakan oleh semua pihak terlebih pada tingkat pemilihan nasional.

DAFTAR PUSTAKA

Abidin, Z., & Khairudin, F. (2017). Penafsiran Ayat-Ayat Amanah dalam Al-Qur'an. Riau. *Jurnal Ilmu Al-Qur'an Dan Keislaman*.

Al-Qur'an dan Terjemahnya. (2019). Kementrian Agama RI.

Ardilla, M., & Asrinaldi, A. (2019). Budaya Merantau Masyarakat Dan Permasalahan Pendaftaran Pemilih Pada Pilkada Di Sumatera Barat: *Jurnal Antropologi: Isu-Isu Sosial Budaya*, *20*(2), 157.

Arifin, M., & Sajono, H. H. (2013). Analisa Dan Perancangan Sistem E-Voting Pemilu Raya Bem (Pemira-Bem) Di Universitas Muria Kudus. *Simetris : Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, *3*(1), 17. https://doi.org/10.24176/simet.v3i1.82.

Dang, Q. H. (2015). Secure Hash Standard. *FIBS 180-4 Publication*, *4*(August), 36.

Dewi, I. K. (2021). *Implementasi Algoritma Kriptografi Asimetris Elgamal dalam Proses Enkripsi dan Dekripsi File Teks untuk Meningkatkan Keamanan Data*.

Fajrin, N., Yusuf, M., & Kunci-, K. (2019). Kriptografi Elgamal (Menggunakan Aplikasi Visual Basic).

Kromodimoeljo, S. (2009). Teori dan Aplikasi Kriptograf. In *Zeitschrift fur die Gesamte Hygiene und Ihre Grenzgebiete* (Vol. 30, Issue 6).

Kurniadi, H. (2015). Implementasi Algoritma Kriptografi Elgamal Untuk File Citra 2 Dimensi. *Publikasi Jurnal Skripsi Konsentrasi Rekaya Komputer*, 1–6.

Nisa, L., Indriyani, T., & Ruswiansari, M. (2020). Aplikasi Enkripsi Citra dan Teks Menggunakan Algoritma Diffie-Hellman dan ElGamal. *Jurnal Teknologi Dan Manajemen*, *1*(1), 8–17.

Purnomo, A. (2018). *Algoritma RSA dan Elgamal sebagai Algoritma Tambahan untuk Mengatasi Kelemahan Algoritma One Time Pad pada Skema Three Pass Protocol*.

Quthb, S. (2000). *Al-Mukminun-Indonesia* (p. 340).

Sebastian, A. (2007). *Implementasi dan perbandingan performa algoritma hash sha-1, sha-256, dan sha-512*. 1–18.

Shita, R. T., & Hin, L. L. (2018). Implementasi Algoritma Kriptografi Aes 128Bit Dan Elgamal Untuk Pengamanan E-Mail Pada Bandara Internasional Sultan Mahmud Baharuddin II. Jakarta: *Budi Luhur Information Technology*.

Suganda, F. S. (2019). *Rancang Bangun Aplikasi E-Voting Presiden Mahasiswa Universitas Nusa Putra Berbasis Android Menggunakan Metode Geofence*

Design And Development Of Android Based E-Voting Application For President Of Nusa Putra Students Using Geofence Method. *1*.

Sulastri, S., & Putri, R. D. M. (2018). Implementasi Enkripsi Data Secure Hash Algorithm (SHA-256) dan Message Digest Algorithm (MD5) pada Proses Pengamanan Kata Sandi Sistem Penjadwalan Karyawan. *Jurnal Teknik Elektro*, *10*(2), 70–74.

LAMPIRAN

Lampiran 1 Bilangan Biner Pada 116 NIM

No.	NIM	Karakter dalam Biner ASCII
1	18610001	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110001
2	18610002	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110010
3	18610003	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110011
4	18610004	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110100
5	18610005	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110101
6	18610006	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110110
7	18610007	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00110111
8	18610008	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00111000
9	18610009	00110001 00111000 00110110 00110001 00110000 00110000 00110000 00111001
10	18610010	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110000
11	18610011	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110001
12	18610012	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110010
13	18610013	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110011
14	18610014	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110100
15	18610015	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110101
16	18610016	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110110
17	18610017	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00110111
18	18610018	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00111000

No.	NIM	Karakter dalam Biner ASCII
19	18610019	00110001 00111000 00110110 00110001 00110000 00110000 00110001 00111001
20	18610020	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110000
21	18610021	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110001
22	18610022	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110010
23	18610023	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110011
24	18610024	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110100
25	18610025	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110101
26	18610026	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110110
27	18610027	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00110111
28	18610028	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00111000
29	18610029	00110001 00111000 00110110 00110001 00110000 00110000 00110010 00111001
30	18610030	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110000
31	18610031	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110001
32	18610032	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110010
33	18610033	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110011
34	18610034	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110100
35	18610035	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110101
36	18610036	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110110
37	18610037	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00110111
38	18610038	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00111000

No.	NIM	Karakter dalam Biner ASCII
39	18610039	00110001 00111000 00110110 00110001 00110000 00110000 00110011 00111001
40	18610040	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110000
41	18610041	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110001
42	18610042	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110010
43	18610043	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110011
44	18610044	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110100
45	18610045	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110101
46	18610046	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110110
47	18610047	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00110111
48	18610048	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00111000
49	18610049	00110001 00111000 00110110 00110001 00110000 00110000 00110100 00111001
50	18610050	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110000
51	18610051	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110001
52	18610052	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110010
53	18610053	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110011
54	18610054	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110100
55	18610055	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110101
56	18610056	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110110
57	18610057	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00110111
58	18610058	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00111000

No.	NIM	Karakter dalam Biner ASCII
59	18610059	00110001 00111000 00110110 00110001 00110000 00110000 00110101 00111001
60	18610060	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00110000
61	18610061	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00110001
62	18610062	00110001 00111000 00110110 00110001 00110000 00110000 00110110 0011001
63	18610063	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00110011
64	18610064	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00110100
65	18610065	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00110101
66	18610066	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00110110
67	18610067	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00110111
68	18610068	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00111000
69	18610069	00110001 00111000 00110110 00110001 00110000 00110000 00110110 00111001
70	18610070	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110000
71	18610071	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110001
72	18610072	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110010
73	18610073	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110011
74	18610074	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110100
75	18610075	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110101
76	18610076	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110110
77	18610077	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00110111
78	18610078	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00111000

No.	NIM	Karakter dalam Biner ASCII
79	18610079	00110001 00111000 00110110 00110001 00110000 00110000 00110111 00111001
80	18610080	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110000
81	18610081	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110001
82	18610082	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110010
83	18610083	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110011
84	18610084	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110100
85	18610085	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110101
86	18610086	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110110
87	18610087	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00110111
88	18610088	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00111000
89	18610089	00110001 00111000 00110110 00110001 00110000 00110000 00111000 00111001
90	18610090	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00110000
91	18610091	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00110001
92	18610092	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00110010
93	18610093	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00110011
94	18610094	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00110100
95	18610095	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00110101
96	18610096	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00110110
97	18610097	00110001 00111000 00110110 00110001 00110000 00110000 00111001 0011011
98	18610098	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00111000

No.	NIM	Karakter dalam Biner ASCII
99	18610099	00110001 00111000 00110110 00110001 00110000 00110000 00111001 00111001
100	18610100	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110000
101	18610101	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110001
102	18610102	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110010
103	18610103	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110011
104	18610104	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110100
105	18610105	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110101
106	18610106	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110110
107	18610107	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00110111
108	18610108	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00111000
109	18610109	00110001 00111000 00110110 00110001 00110000 00110001 00110000 00111001
110	18610110	00110001 00111000 00110110 00110001 00110000 00110001 00110001 00110000
111	18610111	00110001 00111000 00110110 00110001 00110000 00110001 00110001 00110001
112	18610112	00110001 00111000 00110110 00110001 00110000 00110001 00110001 00110010
113	18610113	00110001 00111000 00110110 00110001 00110000 00110001 00110001 00110011
114	18610114	00110001 00111000 00110110 00110001 00110000 00110001 00110001 00110100
115	18610115	00110001 00111000 00110110 00110001 00110000 00110001 00110001 00110101
116	18610116	00110001 00111000 00110110 00110001 00110000 00110001 00110001 00110110

Lampiran 2 Hasil Pada Proses Fungsi Hash SHA-256

NO.	NIM	Message Digest
1	18610001	$271fff57b6c0749d622563e127d1ada2$ $1a01fbc4c406eb6ae47af1080112a3ec$
2	18610002	$0996d7bd977fb60fd9f1fcdff1add029$ $b57db20719c7a215d7da03f7102eb22c$
3	18610003	$32fc038613cf7a582f6b39dbe8d6001e$ $b806671d61f006568bb1c69c118a593e$
4	18610004	$6fa1640570cbd1f6ba0036453adec924$ $d857c9738daf7cc0feaf4b52b249b062$
5	18610005	$8991bd4ea8fb3887fd23216b8e0dca04$ $bd90e6d03d3052f0d7c49e66d1c5360e$
6	18610006	$dc98c687dbde8acad0ca5b41053167f2$ $861133beaa6f4ffbe4f55a2b9fece9c1$
7	18610007	$4b7edf067970d131c3f15b120c2e6b3b$ $617a6e7f1a97e905e14e06a56298ad68$
8	18610008	$655cfb9fcebbccb0199a399922debd6d$ $fcf521677c633392adf0e04e05bdf135$
9	18610009	$debfc45d8f67510801b04010f201bff0$ $a4300eb00175b15607547d28f0e32930$
10	18610010	$1680514c7e6ec664549c178682b879de$ $23f12bf17a31d126387e2bdd5a54d812$
11	18610011	$009f4ce13757c571fa45726fa68037dc$ $de20c91111086a458446f839cf1197c9$
12	18610012	$083d3826ab888bfb1ba30deab5d85667$ $42f9ffbbe9a1bb20213464df27cb4ce0$
13	18610013	$ef4c364a06b157b93a0826989eef7ebe$ $810db079bfeede605739d6bcf0987c4d$
14	18610014	$1e44558d0e4d7f1478f43e1abc47a341$ $8dc2f62709db622b451c965a5b3d6c25$
15	18610015	$dc3796c73f82220a7bc8adc98df29fd0$ $f5180ba21a71e8ec67dc9f4745c8d5b1$
16	18610016	$4656614e5e5a623ca12678886bbff88b$ $0baf39471e37ccfe797690ba4fbb0dd1$
17	18610017	$4b204e33ac7aedacf9d3eadc9db25296$ $db4cfa862cd0ebdc7c821c2968a2cd97$
18	18610018	$9c20ffd253d3bf33b768d9e697d9a80f$ $6ba58a202c639cf79b6e35411f1a328e$
19	18610019	$baf627e60df5eb418c7c94a0d4d58e34$ $8d04bf3839957d20906d8df75e7c053c$

NO.	NIM	Message Digest
20	18610020	$baf627e60df5eb418c7c94a0d4d58e34$ $8d04bf3839957d20906d8df75e7c053c$
21	18610021	$9d5a3054d05fc0a20e05952d4a26e215$ $7195c3c7f93b2bcff3eccb349528840d$
22	18610022	$39c050345e0dcff58770b8d3a5d9b017$ $b5f4d9181678e4f62be1bbbcafadc760$
23	18610023	$68c561a33c58bfcbafa33e137356bbd5$ $d5ab605b22b77b5880407d319337a934$
24	18610024	$4acbd56ac16ab7f0e3af1354521f5665$ $802c78cd518264d5f9c7346a6ed66cd8$
25	18610025	$504fddb20f5efc454f7ac85cfc826f4e$ $cc57c095550305b69569d9629dbb3323$
26	18610026	$ebf08bd5b20531b3d6b67bb72c8119e4$ $c384d60867c8710c5b82f201e8d72e7e$
27	18610027	$6453b656033e16f1ff1c3b0ae5034912$ $5e6e5768c0719aaba1b360fcdc46da35$
28	18610028	$b6e9f754436c844bb68b64fba9aa27ff$ $5629077669c287008ee2bc81c1f29771$
29	18610029	$d8a9da824da6356589562be6c610ca0e$ $1e757d1c54b342a11a786d95497c6034$
30	18610030	$1e856abd43ed1c760668dff239489006$ $c9d28a7d90dd27ad9b9a66277cc744eb$
31	18610031	$6a00e54e6726bb427b0d9c9c06d76755$ $4cf11962ea780666003e65a487cd4596$
32	18610032	$25ed1f2535d40cc600aaeebec659bba9$ $3e036ddb3eef32e49bd30fbf858e2c56$
33	18610033	$8cf9f0e2090567f38ce6fa2b910685c4$ $4f2051e156742b143e4003be01854e22$
34	18610034	$1031c31f0553c2f32249e33df5d17b38$ $f5971015493fc177d3846e7e53971387$
35	18610035	$f1d1ff9e3bdc52111704e913a3f8094b$ $4299abe68a4ab5f3320d0a792f2c6328$
36	18610036	$c0476e68ea18d38a2d131afbf25f1361$ $a8d00a032eceb87a0b596ff7e3588a3d$
37	18610037	$cf3429ec746fb137b13a1384e5cb3674$ $ac056728d09a27ee4f87b36e0b602268$
38	18610038	$f11815ed7f4153c54193baef02c27ec3$ $21595a567c2233bc96688530f3d0873d$
39	18610039	$c5241672766da6bf02e6de2267bf3bf6$ $1aa88423d7d367a53fb21d0a49049cbf$

NO.	NIM	Message Digest
40	18610040	$f0958629c3adbd63812e7cd39af36089$ $af7a6723d29d4a28373b10e1fca0cddb$
41	18610041	$c2e790065189918f959010bbddb3d10b$ $be8e62df0d7d608823bbe77c9b0673fd$
42	18610042	$ef5b3cd6f9d56f43d24f09192d4c7133$ $0e96c2d36653595a65c506468d3d1f13$
43	18610043	$308a869ffdbff115bfa56ba3e72fbe87$ $0ebbd3ed3e3ade988a3c5f0f9b740cad$
44	18610044	$777faa04b23054ffadfe14682acae762$ $5b5b76cd9084bfc4d2d1fa7c9ff42209$
45	18610045	$32120025e06e35ed04d598bd46efea49$ $6c63c893a1386316537495c7a0c295ef$
46	18610046	$0e52d9db3ec1993aaf52d200ce5748b9$ $a175d05780d0ce5f6d5d7fadbaef90a7$
47	18610047	$31e5167a15247c4539eee8ca2a403ad2$ $e463a70c70b435d5f26111f25f051d30$
48	18610048	$2173062037e738716e2b79aa0736cb3c$ $c4d15d1f864aaab77f3370faf6ff2946$
49	18610049	$11dc581f7d32aa44dcb925bd424dabde$ $874ccc6f1de261e0e670d13417daf646$
50	18610050	$e02548e5c3bf2f8ab9aa7f6f38c303cc$ $2be2a51649e967d85b193b491868fcea$
51	18610051	$fd3b0c9ef87f432ab1d47831b5ab12c$ $30e27c9694d3b6509faa85809bcce5025$
52	18610052	$4fe2e68a6b811f33870d686eb90e045$ $e16699dcd7e649485adbfe093260b5f31$
53	18610053	$b28445fb4284ed00e6e09bcc535e9a7a$ $bc81f417feec5e695ede20fdce512c3f$
54	18610054	$c21ece1a007e67801c1ec999e9ed6056$ $b0e8c0b14a280c94ff364dbfc5c11072$
55	18610055	$8b1eb36324b65604d2f89b660a031e6a$ $8ba5e5a00af55b126e55a1f6ac4c4ffd$
56	18610056	$45baeb84e2ca6d9837b10c5032ab56f6$ $04da740f9a1fa3cb590076dbdd53c559$
57	18610057	$672e45205aa15463d7ba1221b2812c45$ $c1c647683069874ebda2229e7c8cb958$
58	18610058	$85e5c4e6620fc612a8a831c04724729c$ $cb4ee77ea42bea49cc3fcbc354d45187$
59	18610059	$2ec05e8ac5a50ff953dc9b1f2d4f0487$ $5070e051819b00a940607aa5d7fc0655$

NO.	NIM	Message Digest
60	18610060	06e849190729f4ac348c8a3304e9cc09 b2f80222964b95aad69cd004b2165dec
61	18610061	e48f42abd30c35345aa850cd34d5b6d 9a8878e28b5eed9efa373c4aa36cbb1d3
62	18610062	17ca22c86d34d5cf52c43d1a4985e80a 54868c752566d6065d127ff0645f11b6
63	18610063	1d274dba0c2bd325aab71291fd851ec1 d5e186f7f915dec8988a05533fa18c22
64	18610064	5bd9054b51f11358ed2b20abcc4d3b94 cf6ca7be85ff06e44af9490df415bdda
65	18610065	554d7b73060c6ab6d7e2373c96f81a01 e28b117d027d8feac4d37a51f05d6aa9
66	18610066	748b2a4b54709ea053a3a5fd766665b5 8a365e06560efba68c6b447f6bcff2e7
67	18610067	fd53beaccca6db7f347f4e1d62f36279 d604aa3cbf6a5907f4faac1309526053
68	18610068	6dd7ed7fa772b3db27a92bcf5830df8a 590dff4aabacd59fc90c058570fba4de
69	18610069	d96994bb2fcd072e6bbae0d8f38fa86d acb2bf7122670b5dfdbe8dfe3bb46c09
70	18610070	05dd7970b8b05adf227712e6d7efcf45 ebbf55db0bae93311233eec30c966e30
71	18610071	625db7dcdbd42c3f4c553ca786d64275 b8ce820de7fd6f439faba26acb469fa6
72	18610072	736b3cf5c2350e563c2f0ae744e30d24 528a4f3d26c8731b7b72cb73cb8c0de4
73	18610073	9c75391fe0366efac1154d9819387903 49a342b38b371c1e36c258b5abee248e
74	18610074	ac701a9d71e40a91e396a742605cc54 fc64c7346cdb24e98d4c1aa39ccefbc72
75	18610075	70dd006f556e0f0cb4d21ff9028b1ba3 976fcec5c14fd14355fa54667a88b804
76	18610076	96bbfbd5571e108b8248db2a56de4330 b98d78dd8f13e00abac5b1e355c9172d
77	18610077	7d0f89a26702660fe5e324df14b69747 42973cf01470e4bca970b05f94d0e2e8
78	18610078	1799b7f40ed9bd9cd7f7392cdf4b37b9 b6d83f7bb1e190f8cf8cde6c7c317f7b
79	18610079	15f3f2cb9f0d8d6ce666a479a8b6bb51 517e0e9f917e5648bcbb7f789eefd8a1

NO.	NIM	Message Digest
80	18610080	411e8768e3f1b043c81360c0452d68c2 b00eeaca6ce0033ae618b7af1fdd4e88
81	18610081	9e6bc07d78744b0271d620b3eec72b1e df70d68a59d6b56447bad22a589a164f
82	18610082	416675d0ece48dccdf9b861606056606 9ac1f8338d04d27cc01b0215107063d7
83	18610083	39a5ffb24fa26b6286cbb92831616421 0cfffd9ce851c40965bb6d8f4afa3b02
84	18610084	33dc47a53a940f0c99011aa362340543 b21f4ed55753598b7d6e9c693641c9bc
85	18610085	d860f2f6aea61e91d6a9180e4feea18f 3023ef134952b1ddf497d082d999d22e
86	18610086	6642f3108b4fc76ce7cee54dca4eda49 065a45682d7f97910711e6f0265f539d
87	18610087	c91b9d9366d7588d289443910c62bdc4 d88796f550700fd29617b78da83413da
88	18610088	35ce0bdbcd43092faf47ae9c393fe84f 1cf8facc52bfe8f24604e6f4b97ad6e8
89	18610089	3c94228cfa679ffdbc0da1f4a1868751 9cea2bb824f95aa6e1a08e916224344f
90	18610090	b5d6a4ef54f2c6b87cd92f90e4c7c41e 06912c09a8452c87c600e037b170d5b5
91	18610091	128e74c710f42cdfae28d9c7eafca039 13dd561cb01fe9b16e14f4859d15aa20
92	18610092	3d79e8677cca84c382017fc222a57ac3 01b8a4268dc67f09261cba803a60b33e
93	18610093	0619cc87b10b35e8f09e96ec0f2cf1af 23cecd163cf97491636b6e4bfddf537d
94	18610094	ba743163ccfd7e39c13924853e0c146d a2488bcc49de1ef449d944ef98835d2d
95	18610095	bf09f0ac1253be831f9327959f665d2 60dcf3ff8fc81f23ecc92b552acbee3cd
96	18610096	4b02afcbb3e7943053df0e1182bec430 214f314f5e295cfb893eefbeb7af8fa5
97	18610097	f0198a31ab1146b91c994ea0a1acc9eb dd9bd1c4722ffc9980de815babe53084
98	18610098	70d4824fe3292b3ed969c9b1f244769d 3a25d91728d2c356c0652060a4f4060b
99	18610099	c08ae76f9591650f9b65de3cd523852b 524f091fc0495079cf36a914ab089f69

NO.	NIM	Message Digest
100	18610100	3d0ca0db6c01c7665a34f9c1304900c9 8795ccb9673f281381fadcabd668a6b8
101	18610101	77c1162e8de529b84191eba723728a1e dbd32e95bc18f55559811d30542ab019
102	18610102	cd6f79b860dbc51b3e00f1790426 9ff6 85b640ee41ce05f3fde707494b82831f
103	18610103	6120bc617530bd58a36572be9bad7084 ee96fc20863cf04ce4601e168bd14723
104	18610104	ad8f92db372bfebf74bbc5d230a2f30a fd9cad3a63c2205b3dd36db3640a2075
105	18610105	0b400cfaee30f2b09b129b0b9102fce1d 0bd48062c85fad25d07ef32ca3cd3a0
106	18610106	68a3d07370d72b89fc7473b67a53fd90 ff5b4aded9c4b45a030d09e69957206b
107	18610107	79e1bc43198e40e08aea81dfb234c3262 075263258df26448cca999793b15840
108	18610108	1b3877780572b74fa35e4a10dfb5c296 ee358fa67d0bf970aadd3dca9e1336d5
109	18610109	cac49fbf55126992105c560f69f7828c 85832cf576be48e0090ffb034a2615f3
110	18610110	c725011aae89a50f47e6f9572e58f753 fd7b2ee48e30b49642079aae56bd343d
111	18610111	1a7c6d04cf2c0b8778690f91d972a02a 46ffb3fcaa278fa233f03690d982e212
112	18610112	77e42bedd0f4c4e1c551903df1063325 740826cb28c50ce6f884b9f5071f3d06
113	18610113	2d82a611cb62b5600882aba361429cf2 21b8e7bcf9a600804a1b68f1d026f07c
114	18610114	7100d1cfb4c71dfdd0fa5163e03fe929 c7aa603dc734dbe1d00f92ebc5c314ce
115	18610115	3f88ec64718c1d1d83bccf69def24001 b43563941287abeb737ad0bca22cc96e
116	18610116	4c9d26366551e2e1d702fd9f7b64263b 780ea93c459805e6f34ac97bd358c6ad

Lampiran 3 Hasil Pembangkitan Kunci Algoritma ElGamal

NO	NIM	p	g	x	k	(y, g, p)
1	18610001	1553	3	61	165	(695,3,1553)
2	18610002	2473	5	62	266	(1922,5,2473)
3	18610003	4219	2	63	365	(162,2,4219)
4	18610004	1399	13	64	466	(1280,13,1399)
5	18610005	3191	11	65	565	(2127,11,3191)
6	18610006	2741	2	66	666	(1188,2,2741)
7	18610007	3041	3	67	765	(2137,3,3041)
8	18610008	3001	14	68	866	(4,14,3001)
9	18610009	2851	2	69	965	(2367,2,2851)
10	18610010	3089	3	61	166	(594,3,3089)
11	18610011	4909	6	62	265	(2354,6,4909)
12	18610012	2161	23	63	366	(869,23,2161)
13	18610013	3343	5	64	465	(2557,5,3343)
14	18610014	1433	3	65	566	(848,3,1433)
15	18610015	3671	13	66	665	(2922,13,3671)
16	18610016	3191	11	67	766	(2087,13,3191)
17	18610017	4019	2	68	865	(3145,2,4019)
18	18610018	4663	3	69	966	(383,3,4663)
19	18610019	3697	5	61	165	(2484,5,3697)
20	18610020	3923	2	62	266	(3641,2,3923)
21	18610021	1091	2	63	365	(789,2,1091)
22	18610022	4409	3	64	466	(3590,3,4409)
23	18610023	3517	2	65	565	(2918,2,3517)
24	18610024	4337	3	66	666	(3736,3,4337)
25	18610025	1787	2	67	765	(1736,2,1787)
26	18610026	1571	2	68	866	(1078,2,1571)
27	18610027	4423	3	69	965	(2857,3,4423)
28	18610028	3331	3	61	166	(2277,3,3331)
29	18610029	2131	2	62	265	(1778,2,2131)
30	18610030	2677	2	63	366	(922,2,2677)

NO	NIM	p	g	x	k	(y, g, p)
31	18610031	4969	11	64	465	(1977,11,4969)
32	18610032	4013	2	65	566	(3443,2,4013)
33	18610033	1181	7	66	665	(948,7,1181)
34	18610034	2111	7	67	766	(1946,7,2111)
35	18610035	4663	3	68	865	(1682,3,4663)
36	18610036	2281	7	69	966	(1588,7,2281)
37	18610037	4483	2	61	165	(4068,2,4483)
38	18610038	2833	5	62	266	(1855,5,2833)
39	18610039	2477	2	63	365	(1676,2,2477)
40	18610040	1789	6	64	466	(1251,6,1789)
41	18610041	1571	2	65	565	(1313,2,1571)
42	18610042	1249	7	66	666	(807,7,1249)
43	18610043	1487	5	67	765	(919,5,1487)
44	18610044	2423	5	68	866	(1921,5,2423)
45	18610045	2713	5	69	965	(575,5,2713)
46	18610046	3391	3	61	166	(1022,3,3391)
47	18610047	4409	3	62	265	(4318,3,4409)
48	18610048	1277	2	63	366	(18,2,1277)
49	18610049	2659	2	64	465	(437,2,2659)
50	18610050	3457	7	65	566	(818,7,3457)
51	18610051	1373	2	66	665	(135,2,1373)
52	18610052	3167	5	67	766	(816,5,3167)
53	18610053	4703	5	68	865	(2614,5,4703)
54	18610054	3911	13	69	966	(1928,13,3911)
55	18610055	1783	10	61	165	(1532,10,1783)
56	18610056	1049	3	62	266	(595,3,1049)
57	18610057	3257	3	63	365	(2501,3,3257)
58	18610058	2621	2	64	466	(1536,2,2621)
59	18610059	2129	3	65	565	(393,2,2129)
60	18610060	1019	2	66	666	(361,2,1019)
61	18610061	3823	3	67	765	(2973,3,3823)

NO	NIM	p	g	x	k	(y, g, p)
62	18610062	1361	3	68	866	(268,3,1361)
63	18610063	4019	2	69	965	(2271,2,4019)
64	18610064	1511	11	61	166	(709,11,1511)
65	18610065	2557	2	62	265	(289,2,2557)
66	18610066	1439	7	63	366	(792,7,1439)
67	18610067	4397	2	64	465	(630,2,4397)
68	18610068	2957	2	65	566	(1240,2,2957)
69	18610069	3259	3	66	665	(1353,2,3259)
70	18610070	3701	2	67	766	(1158,2,3701)
71	18610071	2857	11	68	865	(877,11,2857)
72	18610072	3571	2	69	966	(1277,2,3571)
73	18610073	3529	17	61	165	(2064,17,3529)
74	18610074	4751	19	62	266	(1333,19,4751)
75	18610075	4243	2	63	365	(1782,2,4243)
76	18610076	3967	6	64	466	(2050,6,3967)
77	18610077	3709	2	65	565	(1419,2,3709)
78	18610078	2341	7	66	666	(2042,7,2341)
79	18610079	3163	3	67	765	(1420,3,3163)
80	18610080	4591	11	68	866	(1970,11,4591)
81	18610081	3917	2	69	965	(1903,2,3917)
82	18610082	2957	2	61	166	(1552,2,2957)
83	18610083	3847	5	62	265	(3010,5,3847)
84	18610084	4817	3	63	366	(2040,3,4817)
85	18610085	3253	2	64	465	(1021,2,3253)
86	18610086	4363	2	65	566	(3380,2,4363)
87	18610087	1091	2	66	665	(857,2,1091)
88	18610088	2503	3	67	766	(1725,3,2503)
89	18610089	3821	3	68	865	(2194,3,3821)
90	18610090	2777	3	69	966	(255,3,2777)
91	18610091	4679	11	61	165	(1853,11,4679)
92	18610092	2381	3	62	266	(1876,3,2381)

NO	NIM	p	g	x	k	(y, g, p)
93	18610093	3917	2	63	365	(1315,2,3917)
94	18610094	1861	2	64	466	(559,2,1861)
95	18610095	2879	7	65	565	(2226,7,2879)
96	18610096	4363	2	66	666	(2397,2,4363)
97	18610097	4523	5	67	765	(2461,5,4523)
98	18610098	3023	5	68	866	(1521,5,3023)
99	18610099	1373	2	69	965	(1080,2,1373)
100	18610100	4663	3	61	166	(3054,3,4663)
101	18610101	4751	19	62	265	(1333,19,4751)
102	18610102	1871	14	63	365	(268,14,1871)
103	18610103	2383	5	64	465	(852,5,2383)
104	18610104	4027	3	65	565	(2689,3,4027)
105	18610105	3907	2	66	665	(805,2,3097)
106	18610106	1637	2	67	765	(111,2,1637)
107	18610107	4787	2	68	865	(3589,2,4787)
108	18610108	4973	2	69	965	(311,2,4973)
109	18610109	3389	3	61	165	(39,3,3389)
110	18610110	1433	3	62	265	(456,3,1433)
111	18610111	1489	14	63	365	(938,14,1489)
112	18610112	1297	10	64	465	(76,10,1297)
113	18610113	2621	2	65	565	(451,2,2621)
114	18610114	4951	6	66	665	(2007,6,4951)
115	18610115	3929	3	67	765	(3254,3,3929)
116	18610116	2293	2	68	865	(1428,2,2293)

Lampiran 4 Hasil Enkripsi dengan Algoritma ElGamal

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
1.	18610001	0A1	1235,968	1235,1052	1235,1247
		0B1	599,311	599,1010	599,91
2.	18610002	0A2	2245,107	2245,1639	2245,1451
		0B2	1333,395	1333,234	1333,1751
3.	18610003	0A3	84,4200	84,3578	84,2353
		0B3	168, 1141	168,3151	168,1476
4.	18610036	0A4	508,1157	508,1013	508,1370
		0B4	1008,818	1008,775	1008,653
5.	18610005	0A5	2032,939	2032,1471	2032,2034
		0B5	15,2878	15,1564	15,2513
6.	18610006	0A6	2354,389	2354,2297	2354,95
		0B6	1967,1644	1967,890	1967,479
7.	18610007	0A7	2581,2796	2581,3026	2581,923
		0B7	1661,2528	1661,435	1661,1883
8	18610008	0A8	67,932	67,1012	67,87
		0B8	938,727	938,2125	928,1805
9	18610009	0A9	1198,1903	1198,1840	1198,122
		0B9	2396,2672	2396,823	2396,322
10	18610010	1A0	141,2954	141,3036	141,246
		1B0	423,124	423,1680	423941
11	18610011	1A1	3538 , 2653	3538 , 2818	3538 , 2653
		1B1	1592 , 914	1592 , 630	1592 , 914
12	18610012	1A2	567 , 567	567 , 1987	567 , 1196
		1B2	75 , 15	75 , 2093	75 , 2044
13	18610013	1A3	1000 , 2725	1000 , 3069	1000 , 2768
		1B3	1657 , 1013	1657 , 1228	1657 , 645
14	18610014	1A4	689 , 1367	689 , 1053	689 , 1129
		1B4	634 , 1352	634 , 739	634 , 148
15	18610015	1A5	1013 , 443	1013 , 1187	1013 , 629
		1B5	2156 , 2254	2156 , 3036	2156 , 2438

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
16	18610016	1A6	730 , 1664	730 , 1491	730 , 3006
		1B6	1648 , 960	1648 , 2856	1648 , 16
17	18610017	1A7	3317 , 572	3317 , 1661	3317 , 478
		1B7	2615 , 2447	2615 , 3460	2615 , 204
18	18610018	1A8	360 , 4180	360 , 2690	360 , 4111
		1B8	1080 , 1531	1080 , 1967	1080 , 3082
19	18610019	1A9	3639 , 3167	3639 , 2164	3639 , 817
		1B9	3407 , 3309	3407 , 2269	3407 , 3472
20	18610020	2A0	654 , 1096	654 , 2994	654 , 1366
		2B0	1308 , 845	1308 , 1900	1308 , 3165
21	18610021	2A1	307 , 1003	307 , 322	307 , 612
		2B1	614 , 392	614 , 692	614 , 646
22	18610022	2A2	3182 , 3104	3182 , 508	3182 , 3104
		2B2	728 , 1817	728 , 1693	728 , 1817
23	18610023	2A3	183 , 3235	183 , 2447	183 , 2948
		2B3	366 , 102	366 , 416	366 , 3199
24	18610024	2A4	1403 , 2311	1403 , 3438	1403 , 1883
		2B4	4209 , 3266	4209 , 1015	4209 , 274
25	18610025	2A5	57 , 791	57 , 1207	57 , 1589
		2B5	114 , 760	114 , 1718	114 , 1163
26	18610026	2A6	593 , 1124	593 , 1147	593 , 397
		2B6	1186 , 431	1186 , 1323	1186 , 654
27	18610027	2A7	119 , 1237	119 , 2935	119 , 1803
		2B7	357 , 132	357 , 705	357 , 2799
28	18610028	2A8	1755 , 3041	1755 , 2954	1755 , 2340
		2B8	1934 , 2539	1934 , 3085	1934 , 1911
29	18610029	2A9	2087 , 1758	2087 , 1433	2087 , 896
		2B9	2043 , 1678	2043 , 1874	2043 , 1231
30	18610030	3A0	506 , 2270	506 , 951	506 , 1979
		3B0	1012 , 2203	1012 , 2536	1012 , 1601
31	18610031	3A1	1393 , 594	1393 , 3680	1393 , 863

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
		3B1	416 , 1654	416 , 679	416 , 1784
32	18610032	3A2	852 , 3243	852 , 3189	852 , 1527
		3B2	1704 , 1483	1704 , 1211	1704 , 431
33	18610033	3A3	651 , 570	651 , 657	651 , 570
		3B3	1014 , 643	1014 , 1110	1014 , 643
34	18610034	3A4	622 , 1215	622 , 1176	622 , 1363
		3B4	132 , 70	132 , 1084	132 , 982
35	18610035	3A5	2151 , 4553	2151 , 1597	2151 , 800
		3B5	1790 , 1500	1790 , 844	1790 , 2656
36	18610036	3A6	1665 , 1618	1665 , 1436	1665 , 1579
		3B6	250 , 978	250 , 1534	250 , 633
37	18610037	3A7	3687 , 323	3687 , 1906	3687 , 3337
		3B7	2891 , 445	2891 , 1367	2891 , 392
38	18610038	3A8	707 , 1611	707 , 1220	707 , 1269
		3B8	702 , 2423	702 , 136	702 , 2605
39	18610039	3A9	660 , 1977	660 , 2374	660 , 24
		3B9	1320 , 1703	1320 , 164	1320 , 592
40	18610040	4A0	1128 , 902	1128 , 233	1128 , 695
		4B0	1401 , 1332	1401 , 1553	1401 , 1780
41	18610041	4A1	262 , 320	262 , 400	262 , 1510
		4B1	1128 , 695	524 , 711	524 , 28
42	18610042	4A2	795 , 346	795 , 1057	795 , 717
		4B2	569 , 695	569 , 738	569 , 332
43	18610043	4A3	1319 , 767	1319 , 587	1319 , 1124
		4B3	647 , 35	647 , 216	647 , 978
44	18610044	4A4	276 , 791	276 , 383	276 , 791
		4B4	1380 , 290	1380 , 1300	1380 , 290
45	18610045	4A5	1609 , 2468	1609 , 372	1609 , 1472
		4B5	2619 , 201	2619 , 2029	2619 , 2657
46	18610046	4A6	2953 , 259	2953 , 703	2953 , 3349
		4B6	2077 , 2372	2077 , 663	2077 , 1159

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
47	18610047	4A7	3345 , 513	3345 , 3948	3345 , 1984
		4B7	1217 , 1816	1217 , 270	1217 , 225
48	18610048	4A8	271 , 948	271 , 1185	271 , 628
		4B8	542 , 463	542 , 735	542 , 1088
49	18610049	4A9	1904 , 1656	1904 , 2070	1904 , 588
		4B9	1149 , 424	1149 , 2379	1149 , 1692
50	18610050	5A0	1646 , 2088	1646 , 1191	1646 , 3326
		5B0	1151 , 226	1151 , 2173	1151 , 9
51	18610051	5A1	449 , 671	449 , 253	449 , 1268
		5B1	898 , 1340	898 , 1306	898 , 928
52	18610052	5A2	1526 , 1163	1526 , 530	1526 , 2113
		5B2	1296 , 2075	1296 , 3062	1296 , 1360
53	18610053	5A3	1671 , 375	1671 , 1436	1671 , 982
		5B3	3652 , 2026	3652 , 2168	3652 , 3813
54	18610054	5A4	2144 , 2841	2144 , 16	2144 , 795
		5B4	495 , 2048	495 , 1960	495 , 3559
55	18610055	5A5	1485 , 970	1485 , 685	1485 , 970
		5B5	586 , 801	586 , 291	586 , 801
56	18610056	5A6	37 , 288	37 , 373	37 , 907
		5B6	111 , 373	111 , 702	111 , 479
57	18610057	5A7	2719 , 1135	2719 , 716	2719 , 1608
		5B7	1643 , 1788	1643 , 2964	1643 , 2470
58	18610058	5A8	694 , 668	694 , 28	694 , 508
		5B8	1388 , 1237	1388 , 403	1388 , 1851
59	18610059	5A9	1103 , 1207	1103 , 2123	1103 , 93
		5B9	1180 , 1713	1180 , 356	1180 , 2093
60	18610060	6A0	583 , 892	583 , 885	583 , 340
		6B0	147 , 8	147 , 123	147 , 460
61	18610061	6A1	3076 , 2458	3076 , 1118	3076 , 2372
		6B1	1582 , 1881	1582 , 2299	1582 , 2344
62	18610062	6A2	975 , 869	975 , 920	975 , 603

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
		6B2	203 , 161	203 , 348	203 , 1006
63	18610063	6A3	801 , 2601	801 , 2461	801 , 447
		6B3	1602 , 2960	1602 , 1385	1602 , 234
64	18610064	6A4	298 , 314	298 , 322	298 , 862
		6B4	256 , 509	256 , 790	256 , 714
65	18610065	6A5	164 , 2017	164 , 1907	164 , 2027
		6B5	328 , 2474	328 , 1035	328 , 250
66	18610066	6A6	1093 , 1331	1093 , 1309	1093 , 1331
		6B6	456 , 804	456 , 503	456 , 804
67	18610067	6A7	3021 , 2431	3021 , 402	3021 , 3046
		6B7	1645 , 1374	1645 , 3145	1645 , 1888
68	18610068	6A8	902 , 1262	902 , 1081	902 , 2842
		6B8	1804 , 627	1804 , 1752	1804 , 2293
69	18610069	6A9	342 , 2291	342 , 1611	342 , 1513
		6B9	1026 , 414	1026 , 506	1026 , 437
70	18610070	7A0	683 , 574	683 , 3370	683 , 3058
		7B0	1366 , 2213	1366 , 435	1366 , 3008
71	18610071	7A1	1825 , 2143	1825 , 1234	1825 , 2117
		7B1	76 , 2362	76 , 2263	76 , 2416
72	18610072	7A2	1575 , 1844	1575 , 1530	1575 , 2001
		7B2	3150 , 1499	3150 , 2513	3150 , 2012
73	18610073	7A3	283 , 1230	283 , 812	283 , 2103
		7B3	1282 , 1369	1282 , 937	1282 , 3451
74	18610074	7A4	2620 , 3863	2620 , 1542	2620 , 3134
		7B4	2270 , 4046	2270 , 3905	2270 , 1493
75	18610075	7A5	3592 , 3946	3592 , 3892	3592 , 1411
		7B5	2941 , 1121	2941 , 3891	2941 , 2546
76	18610076	7A6	905 , 1765	905 , 1004	905 , 651
		7B6	1463 , 346	1463 , 2002	1463 , 1638
77	18610077	7A7	1822 , 2405	1822 , 482	1822 , 2405
		7B7	3644 , 415	3644 , 498	3644 , 415

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
78	18610078	7A8	892 , 2276	892 , 136	892 , 2062
		7B8	1562 , 707	1562 , 2253	1562 , 1486
79	18610079	7A9	1858 , 1676	1858 , 543	1858 , 2082
		7B9	2411 , 1344	2411 , 2198	2411 , 2198
80	18610080	8A0	1946 , 2924	1946 , 3066	1946 , 3818
		8B0	3042 , 3166	3042 , 780	3042 , 1402
81	18610081	8A1	3232 , 463	3232 , 3685	3232 , 1874
		8B1	2547 , 3681	2547 , 841	2547 , 1752
82	18610082	8A2	1214 , 811	1214 , 1839	1214 , 2097
		8B2	2428 , 2234	2428 , 732	2428 , 1361
83	18610083	8A3	2675 , 2844	2675 , 828	2675 , 828
		8B3	1834 , 865	1834 , 2256	1834 , 2093
84	18610084	8A4	2551 , 4307	2551 , 4053	2551 , 2279
		8B4	2836 , 72	2836 , 773	2836 , 755
85	18610085	8A5	1684 , 2186	1684 , 1027	1684 , 1488
		8B5	115 , 348	115 , 2269	115 , 97
86	18610086	8A6	432 , 1008	432 , 1170	432 , 972
		8B6	864 , 3900	864 , 1480	864 , 21
87	18610087	8A7	416 , 582	416 , 169	416 , 143
		8B7	832 , 187	832 , 649	832 , 359
88	18610088	8A8	1138 , 1498	1138 , 1113	1138 , 1498
		8B8	911 , 954	911 , 588	911 , 954
89	18610089	8A9	2404 , 3561	2404 , 517	2404 , 1100
		8B9	3391 , 2921	3391 , 2349	3391 , 2710
90	18610090	9A0	2570 , 551	2570 , 1554	2570 , 464
		9B0	2156 , 1655	2156 , 1624	2156 , 1686
91	18610091	9A1	4266 , 3515	4266 , 889	4266 , 1462
		9B1	136 , 127	136 , 3841	136 , 4624
92	18610092	9A2	1178 , 366	1178 , 668	1178 , 697
		9B2	1153 , 888	1153 , 151	1153 , 403
93	18610093	9A3	1518 , 2119	1518 , 2760	1518 , 659

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
		9B3	3036 , 1498	3036 , 2353	3036 , 928
94	18610094	9A4	61 , 57	61 , 65	61 , 52
		9B4	122 , 226	122 , 1535	122 , 1153
95	18610095	9A5	1694 , 1597	1694 , 1013	1694 , 1889
		9B5	342 , 2236	342 , 2286	342 , 1574
96	18610096	9A6	223 , 917	223 , 4184	223 , 1328
		9B6	446 , 3460	446 , 1710	446 , 2589
97	18610097	9A7	700 , 4071	700 , 2976	700 , 3214
		9B7	3500 , 286	3500 , 4140	3500 , 3450
98	18610098	9A8	457 , 1849	457 , 2851	457 , 968
		9B8	2285 , 939	2285 , 2201	2285 , 127
99	18610099	9A9	326 , 490	326 , 896	326 , 490
		9B9	652 , 595	652 , 978	652 , 595
100	18610100	0A0	1288 , 4309	1288 , 2435	1288 , 4309
		0B0	3864 , 700	3864 , 3294	3864 , 700
101	18610101	0A1	2620 , 2162	2620 , 1542	2620 , 2405
		0B1	2270 , 2840	2270 , 3905	2270 , 3691
102	18610102	0A2	658 , 1306	658 , 911	658 , 269
		0B2	1728 , 131	1728 , 414	1728 , 994
103	18610103	0A3	42 , 1720	42 , 1932	42 , 636
		0B3	210 , 2278	210 , 1345	210 , 931
104	18610104	0A4	3196 , 3373	3196 , 2638	3196 , 1305
		0B4	1534 , 1193	1534 , 1137	1534 , 1628
105	18610105	0A5	12 , 2146	12 , 3720	12 , 1230
		0B5	24 , 636	24 , 2828	24 , 1679
106	18610106	0A6	811 , 313	811 , 799	811 , 966
		0B6	1622 , 366	1622 , 94	1622 , 821
107	18610107	0A7	3839 , 2322	3839 , 2546	3839 , 3259
		0B7	2891 , 4278	2891 , 2292	2891 , 1910
108	18610108	0A8	2437 , 919	2437 , 4249	2437 , 1901
		0B8	4874 , 2348	4874 , 742	4874 , 4397

NO.	NIM	Pesan	a_1, b_1	a_2, b_2	a_3, b_3
109	18610109	0A9	2744 , 1470	2744 , 1567	2744 , 1322
		0B9	1454 , 3106	1454 , 1729	1454 , 723
110	18610110	1A0	372 , 656	372 , 607	372 , 1286
		1B0	1116 , 1072	1116 , 976	1116 , 319
111	18610111	1A1	1322 , 833	1322 , 1105	1322 , 833
		1B1	640 , 1118	640 , 1202	640 , 1118
112	18610112	1A2	1231 , 57	1231 , 605	1231 , 1064
		1B2	637 , 441	637 , 594	637 , 450
113	18610113	1A3	1076 , 617	1076 , 658	1076 , 1605
		1B3	2152 , 441	2152 , 594	2152 , 459
114	18610114	1A4	4107 , 2552	4107 , 152	4107 , 2102
		1B4	4838 , 2530	4838 , 4014	4838 , 462
115	18610115	1A5	2698 , 2796	2698 , 662	2698 , 298
		1B5	236 , 2549	236 , 226	236 , 3158
116	18610116	1A6	1405 , 252	1405 , 1317	1405 , 1588
		1B6	1405 , 252	1405 , 667	1405 , 1588

Lampiran 5 Hasil Dekripsi dengan Algoritma ElGamal

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
1.	18610001	1235,968	0	1235,1052	A	1235,1247	1
		599,311	0	599,1010	B	599,91	1
2.	18610002	2245,107	0	2245,1639	A	2245,1451	2
		1333,395	0	1333,234	B	1333,1751	2
3.	18610003	84,4200	0	84,3578	A	84,2353	3
		168, 1141	0	168,3151	B	168,1476	3
4.	18610036	508,1157	0	508,1013	A	508,1370	4
		1008,818	0	1008,775	B	1008,653	4
5.	18610005	2032,939	0	2032,1471	A	2032,2034	5
		15,2878	0	15,1564	B	15,2513	5
6.	18610006	2354,389	0	2354,2297	A	2354,95	6
		1967,1644	0	1967,890	B	1967,479	6
7.	18610007	2581,2796	0	2581,3026	A	2581,923	7
		1661,2528	0	1661,435	B	1661,1883	7
8	18610008	67,932	0	67,1012	A	67,87	8
		938,727	0	938,2125	B	928,1805	8
9	18610009	1198,1903	0	1198,1840	A	1198,122	9
		2396,2672	0	2396,823	B	2396,322	9
10	18610010	141,2954	1	141,3036	A	141,246	0
		423,124	1	423,1680	B	423941	0
11	18610011	3538 , 2653	1	3538 , 2818	A	3538 , 2653	1
		1592 , 914	1	1592 , 630	B	1592 , 914	1
12	18610012	567 , 567	1	567 , 1987	A	567 , 1196	2
		75 , 15	1	75 , 2093	B	75 , 2044	2
13	18610013	1000 , 2725	1	1000 , 3069	A	1000 , 2768	3
		1657 , 1013	1	1657 , 1228	B	1657 , 645	3

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
14	18610014	689 , 1367	1	689 , 1053	A	689 , 1129	4
		634 , 1352	1	634 , 739	B	634 , 148	4
15	18610015	1013 , 443	1	1013 , 1187	A	1013 , 629	5
		2156 , 2254	1	2156 , 3036	B	2156 , 2438	5
16	18610016	730 , 1664	1	730 , 1491	A	730 , 3006	6
		1648 , 960	1	1648 , 2856	B	1648 , 16	6
17	18610017	3317 , 572	1	3317 , 1661	A	3317 , 478	7
		2615 , 2447	1	2615 , 3460	B	2615 , 204	7
18	18610018	360 , 4180	1	360 , 2690	A	360 , 4111	8
		1080 , 1531	1	1080 , 1967	B	1080 , 3082	8
19	18610019	3639 , 3167	1	3639 , 2164	A	3639 , 817	9
		3407 , 3309	1	3407 , 2269	B	3407 , 3472	9
20	18610020	654 , 1096	2	654 , 2994	A	654 , 1366	0
		1308 , 845	2	1308 , 1900	B	1308 , 3165	0
21	18610021	307 , 1003	2	307 , 322	A	307 , 612	1
		614 , 392	2	614 , 692	B	614 , 646	1
22	18610022	3182 , 3104	2	3182 , 508	A	3182 , 3104	2
		728 , 1817	2	728 , 1693	B	728 , 1817	2
23	18610023	183 , 3235	2	183 , 2447	A	183 , 2948	3
		366 , 102	2	366 , 416	B	366 , 3199	3
24	18610024	1403 , 2311	2	1403 , 3438	A	1403 , 1883	4
		4209 , 3266	2	4209 , 1015	B	4209 , 274	4
25	18610025	57 , 791	2	57 , 1207	A	57 , 1589	5
		114 , 760	2	114 , 1718	B	114 , 1163	5
26	18610026	593 , 1124	2	593 , 1147	A	593 , 397	6
		1186 , 431	2	1186 , 1323	B	1186 , 654	6
27	18610027	119 , 1237	2	119 , 2935	A	119 , 1803	7

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
		357 , 132	2	357 , 705	B	357 , 2799	7
28	18610028	1755 , 3041	2	1755 , 2954	A	1755 , 2340	8
		1934 , 2539	2	1934 , 3085	B	1934 , 1911	8
29	18610029	2087 , 1758	2	2087 , 1433	A	2087 , 896	9
		2043 , 1678	2	2043 , 1874	B	2043 , 1231	9
30	18610030	506 , 2270	2	506 , 951	A	506 , 1979	0
		1012 , 2203	3	1012 , 2536	B	1012 , 1601	0
31	18610031	1393 , 594	3	1393 , 3680	A	1393 , 863	1
		416 , 1654	3	416 , 679	B	416 , 1784	1
32	18610032	852 , 3243	3	852 , 3189	A	852 , 1527	2
		1704 , 1483	3	1704 , 1211	B	1704 , 431	2
33	18610033	651 , 570	3	651 , 657	A	651 , 570	3
		1014 , 643	3	1014 , 1110	B	1014 , 643	3
34	18610034	622 , 1215	3	622 , 1176	A	622 , 1363	4
		132 , 70	3	132 , 1084	B	132 , 982	4
35	18610035	2151 , 4553	3	2151 , 1597	A	2151 , 800	5
		1790 , 1500	3	1790 , 844	B	1790 , 2656	5
36	18610036	1665 , 1618	3	1665 , 1436	A	1665 , 1579	6
		250 , 978	3	250 , 1534	B	250 , 633	6
37	18610037	3687 , 323	3	3687 , 1906	A	3687 , 3337	7
		2891 , 445	3	2891 , 1367	B	2891 , 392	7
38	18610038	707 , 1611	3	707 , 1220	A	707 , 1269	8
		702 , 2423	3	702 , 136	B	702 , 2605	8
39	18610039	660 , 1977	3	660 , 2374	A	660 , 24	9
		1320 , 1703	3	1320 , 164	B	1320 , 592	9
40	18610040	1128 , 902	4	1128 , 233	A	1128 , 695	0
		1401 , 1332	4	1401 , 1553	B	1401 , 1780	0

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
41	18610041	262 , 320	4	262 , 400	A	262 , 1510	1
		1128 , 695	4	524 , 711	B	524 , 28	1
42	18610042	795 , 346	4	795 , 1057	A	795 , 717	2
		569 , 695	4	569 , 738	B	569 , 332	2
43	18610043	1319 , 767	4	1319 , 587	A	1319 , 1124	3
		647 , 35	4	647 , 216	B	647 , 978	3
44	18610044	276 , 791	4	276 , 383	A	276 , 791	4
		1380 , 290	4	1380 , 1300	B	1380 , 290	4
45	18610045	1609 , 2468	4	1609 , 372	A	1609 , 1472	5
		2619 , 201	4	2619 , 2029	B	2619 , 2657	5
46	18610046	2953 , 259	4	2953 , 703	A	2953 , 3349	6
		2077 , 2372	4	2077 , 663	B	2077 , 1159	6
47	18610047	3345 , 513	4	3345 , 3948	A	3345 , 1984	7
		1217 , 1816	4	1217 , 270	B	1217 , 225	7
48	18610048	271 , 948	4	271 , 1185	A	271 , 628	8
		542 , 463	4	542 , 735	B	542 , 1088	8
49	18610049	1904 , 1656	4	1904 , 2070	A	1904 , 588	9
		1149 , 424	4	1149 , 2379	B	1149 , 1692	9
50	18610050	1646 , 2088	4	1646 , 1191	A	1646 , 3326	0
		1151 , 226	5	1151 , 2173	B	1151 , 9	0
51	18610051	449 , 671	5	449 , 253	A	449 , 1268	1
		898 , 1340	5	898 , 1306	B	898 , 928	1
52	18610052	1526 , 1163	5	1526 , 530	A	1526 , 2113	2
		1296 , 2075	5	1296 , 3062	B	1296 , 1360	2
53	18610053	1671 , 375	5	1671 , 1436	A	1671 , 982	3
		3652 , 2026	5	3652 , 2168	B	3652 , 3813	3
54	18610054	2144 , 2841	5	2144 , 16	A	2144 , 795	4

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
		495 , 2048	5	495 , 1960	B	495 , 3559	4
55	18610055	1485 , 970	5	1485 , 685	A	1485 , 970	5
		586 , 801	5	586 , 291	B	586 , 801	5
56	18610056	37 , 288	5	37 , 373	A	37 , 907	6
		111 , 373	5	111 , 702	B	111 , 479	6
57	18610057	2719 , 1135	5	2719 , 716	A	2719 , 1608	7
		1643 , 1788	5	1643 , 2964	B	1643 , 2470	7
58	18610058	694 , 668	5	694 , 28	A	694 , 508	8
		1388 , 1237	5	1388 , 403	B	1388 , 1851	8
59	18610059	1103 , 1207	5	1103 , 2123	A	1103 , 93	9
		1180 , 1713	5	1180 , 356	B	1180 , 2093	9
60	18610060	583 , 892	6	583 , 885	A	583 , 340	0
		147 , 8	6	147 , 123	B	147 , 460	0
61	18610061	3076 , 2458	6	3076 , 1118	A	3076 , 2372	1
		1582 , 1881	6	1582 , 2299	B	1582 , 2344	1
62	18610062	975 , 869	6	975 , 920	A	975 , 603	2
		203 , 161	6	203 , 348	B	203 , 1006	2
63	18610063	801 , 2601	6	801 , 2461	A	801 , 447	3
		1602 , 2960	6	1602 , 1385	B	1602 , 234	3
64	18610064	298 , 314	6	298 , 322	A	298 , 862	4
		256 , 509	6	256 , 790	B	256 , 714	4
65	18610065	164 , 2017	6	164 , 1907	A	164 , 2027	5
		328 , 2474	6	328 , 1035	B	328 , 250	5
66	18610066	1093 , 1331	6	1093 , 1309	A	1093 , 1331	6
		456 , 804	6	456 , 503	B	456 , 804	6
67	18610067	3021 , 2431	6	3021 , 402	A	3021 , 3046	7
		1645 , 1374	6	1645 , 3145	B	1645 , 1888	7

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
68	18610068	902 , 1262	6	902 , 1081	A	902 , 2842	8
		1804 , 627	6	1804 , 1752	B	1804 , 2293	8
69	18610069	342 , 2291	6	342 , 1611	A	342 , 1513	9
		1026 , 414	6	1026 , 506	B	1026 , 437	9
70	18610070	683 , 574	7	683 , 3370	A	683 , 3058	0
		1366 , 2213	7	1366 , 435	B	1366 , 3008	0
71	18610071	1825 , 2143	7	1825 , 1234	A	1825 , 2117	1
		76 , 2362	7	76 , 2263	B	76 , 2416	1
72	18610072	1575 , 1844	7	1575 , 1530	A	1575 , 2001	2
		3150 , 1499	7	3150 , 2513	B	3150 , 2012	2
73	18610073	283 , 1230	7	283 , 812	A	283 , 2103	3
		1282 , 1369	7	1282 , 937	B	1282 , 3451	3
74	18610074	2620 , 3863	7	2620 , 1542	A	2620 , 3134	4
		2270 , 4046	7	2270 , 3905	B	2270 , 1493	4
75	18610075	3592 , 3946	7	3592 , 3892	A	3592 , 1411	5
		2941 , 1121	7	2941 , 3891	B	2941 , 2546	5
76	18610076	905 , 1765	7	905 , 1004	A	905 , 651	6
		1463 , 346	7	1463 , 2002	B	1463 , 1638	6
77	18610077	1822 , 2405	7	1822 , 482	A	1822 , 2405	7
		3644 , 415	7	3644 , 498	B	3644 , 415	7
78	18610078	892 , 2276	7	892 , 136	A	892 , 2062	8
		1562 , 707	7	1562 , 2253	B	1562 , 1486	8
79	18610079	1858 , 1676	7	1858 , 543	A	1858 , 2082	9
		2411 , 1344	7	2411 , 2198	B	2411 , 2198	9
80	18610080	1946 , 2924	7	1946 , 3066	A	1946 , 3818	0
		3042 , 3166	8	3042 , 780	B	3042 , 1402	0
81	18610081	3232 , 463	8	3232 , 3685	A	3232 , 1874	1

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
		2547 , 3681	8	2547 , 841	B	2547 , 1752	1
82	18610082	1214 , 811	8	1214 , 1839	A	1214 , 2097	2
		2428 , 2234	8	2428 , 732	B	2428 , 1361	2
83	18610083	2675 , 2844	8	2675 , 828	A	2675 , 828	3
		1834 , 865	8	1834 , 2256	B	1834 , 2093	3
84	18610084	2551 , 4307	8	2551 , 4053	A	2551 , 2279	4
		2836 , 72	8	2836 , 773	B	2836 , 755	4
85	18610085	1684 , 2186	8	1684 , 1027	A	1684 , 1488	5
		115 , 348	8	115 , 2269	B	115 , 97	5
86	18610086	432 , 1008	8	432 , 1170	A	432 , 972	6
		864 , 3900	8	864 , 1480	B	864 , 21	6
87	18610087	416 , 582	8	416 , 169	A	416 , 143	7
		832 , 187	8	832 , 649	B	832 , 359	7
88	18610088	1138 , 1498	8	1138 , 1113	A	1138 , 1498	8
		911 , 954	8	911 , 588	B	911 , 954	8
89	18610089	2404 , 3561	8	2404 , 517	A	2404 , 1100	9
		3391 , 2921	8	3391 , 2349	B	3391 , 2710	9
90	18610090	2570 , 551	9	2570 , 1554	A	2570 , 464	0
		2156 , 1655	9	2156 , 1624	B	2156 , 1686	0
91	18610091	4266 , 3515	9	4266 , 889	A	4266 , 1462	1
		136 , 127	9	136 , 3841	B	136 , 4624	1
92	18610092	1178 , 366	9	1178 , 668	A	1178 , 697	2
		1153 , 888	9	1153 , 151	B	1153 , 403	2
93	18610093	1518 , 2119	9	1518 , 2760	A	1518 , 659	3
		3036 , 1498	9	3036 , 2353	B	3036 , 928	3
94	18610094	61 , 57	9	61 , 65	A	61 , 52	4
		122 , 226	9	122 , 1535	B	122 , 1153	4

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
95	18610095	1694 , 1597	9	1694 , 1013	A	1694 , 1889	5
		342 , 2236	9	342 , 2286	B	342 , 1574	5
96	18610096	223 , 917	9	223 , 4184	A	223 , 1328	6
		446 , 3460	9	446 , 1710	B	446 , 2589	6
97	18610097	700 , 4071	9	700 , 2976	A	700 , 3214	7
		3500 , 286	9	3500 , 4140	B	3500 , 3450	7
98	18610098	457 , 1849	9	457 , 2851	A	457 , 968	8
		2285 , 939	9	2285 , 2201	B	2285 , 127	8
99	18610099	326 , 490	9	326 , 896	A	326 , 490	9
		652 , 595	9	652 , 978	B	652 , 595	9
100	18610100	1288 , 4309	0	1288 , 2435	A	1288 , 4309	0
		3864 , 700	0	3864 , 3294	B	3864 , 700	0
101	18610101	2620 , 2162	0	2620 , 1542	A	2620 , 2405	1
		2270 , 2840	0	2270 , 3905	B	2270 , 3691	1
102	18610102	658 , 1306	0	658 , 911	A	658 , 269	2
		1728 , 131	0	1728 , 414	B	1728 , 994	2
103	18610103	42 , 1720	0	42 , 1932	A	42 , 636	3
		210 , 2278	0	210 , 1345	B	210 , 931	3
104	18610104	3196 , 3373	0	3196 , 2638	A	3196 , 1305	4
		1534 , 1193	0	1534 , 1137	B	1534 , 1628	4
105	18610105	12 , 2146	0	12 , 3720	A	12 , 1230	5
		24 , 636	0	24 , 2828	B	24 , 1679	5
106	18610106	811 , 313	0	811 , 799	A	811 , 966	6
		1622 , 366	0	1622 , 94	B	1622 , 821	6
107	18610107	3839 , 2322	0	3839 , 2546	A	3839 , 3259	7
		2891 , 4278	0	2891 , 2292	B	2891 , 1910	7
108	18610108	2437 , 919	0	2437 , 4249	A	2437 , 1901	8

No.	NIM	a_1, b_1	m_1	a_2, b_2	m_2	a_3, b_3	m_3
		4874 , 2348	0	4874 , 742	B	4874 , 4397	8
109	18610109	2744 , 1470	0	2744 , 1567	A	2744 , 1322	9
		1454 , 3106	0	1454 , 1729	B	1454 , 723	9
110	18610110	372 , 656	1	372 , 607	A	372 , 1286	0
		1116 , 1072	1	1116 , 976	B	1116 , 319	0
111	18610111	1322 , 833	1	1322 , 1105	A	1322 , 833	1
		640 , 1118	1	640 , 1202	B	640 , 1118	1
112	18610112	1231 , 57	1	1231 , 605	A	1231 , 1064	2
		637 , 441	1	637 , 594	B	637 , 450	2
113	18610113	1076 , 617	1	1076 , 658	A	1076 , 1605	3
		2152 , 441	1	2152 , 594	B	2152 , 459	3
114	18610114	4107 , 2552	1	4107 , 152	A	4107 , 2102	4
		4838 , 2530	1	4838 , 4014	B	4838 , 462	4
115	18610115	2698 , 2796	1	2698 , 662	A	2698 , 298	5
		236 , 2549	1	236 , 226	B	236 , 3158	5
116	18610116	1405 , 252	1	1405 , 1317	A	1405 , 1588	6
		1405 , 252	1	1405 , 667	B	1405 , 1588	6

RIWAYAT HIDUP

Indri Fatikhu Aflikh, dengan nama panggilan Indri, merupakan putri kedua dari Bapak Suwito dan Ibu Ruchayati serta adik dari seorang kakak perempuan Aini Novita Amaliyah. Ia dilahirkan di Kabupaten Malang pada tanggal 19 Februari 2000. Saat ini ia tinggal bersama orang tuanya di Jalan Sunan Ampel no.34 RT 01 RW 04, Dusun Randugembolo, Ardimulyo, Kecamatan Singosari, Kabupaten Malang.

Penulis memulai pendidikan dari TK Al-Masithoh 01 dan lulus pada tahun 2006. Setelah itu penulis menempuh pendidikan sekolah dasar di SD Randuagung 01 dan lulus pada tahun 2012. Penulis kemudian melanjutkan pendidikan sekolah menengah pertama di MTsN Lawang yang saat ini berganti menjadi MTsN 3 Malang dan lulus pada tahun 2015. Pendidikan selanjutnya pada tingkat sekolah menengah atas ditempuh oleh penulis di SMAN 1 Lawang dan merupakan siswa lulusan tahun 2018. Pada tahun yang sama, penulis melanjutkan pendidikannya pada perguruan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang pada Program Studi Matematika, Fakultas Sains dan Teknologi.

Selama menempuh pendidikan, penulis aktif dalam mengikuti beberapa organisasi. Pada tingkat SMA, penulis mengikuti organisasi Pramuka Ambalan Brawijaya-Rajendradewi dengan posisi sebagai sekretaris. Pada tingkat perguruan tinggi, penulis mengikuti Himpunan Mahasiswa Program Studi Matematika sebagai sekretaris I dan II. Penulis juga aktif dalam keanggotaan Senat Mahasiswa Fakultas sebagai anggota Komisi Kesekretariatan dan Kebendaharaan. Penulis juga aktif dalam beberapa komunitas kampus di antaranya Mathematics English Club dan Serambi Matematika Aktif serta komunitas tingkat nasional yakni Lingkar Cendekia dengan posisi sebagai sekretaris.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Indri Fatikhu Aflikh
NIM : 18610036
Fakultas / Program Studi : Sains dan Teknologi / Matematika
Judul Skripsi : Implementasi Algoritma ElGamal dan Fungsi Hash SHA-256 Pada Data *Electronic Voting* (*E-Voting*)

Pembimbing I : Hisyam Fahmi, M.Kom
Pembimbing II : Mohammad Nafie Jauhari, M.Si

No	Tanggal	Hal	Tanda Tangan
1.	24 Februari 2022	Konsultasi BAB I	1.
2.	11 Maret 2022	Konsultasi Kajian Agama Bab I	2.
3.	17 Maret 2022	Konsultasi BAB II dan III	3.
4.	12 April 2022	Konsultasi Revisi BAB II dan III	4.
5.	26 April 2022	Konsultasi Kajian Agama Bab II	5.
6.	02 Mei 2022	Konsultasi Revisi Kajian Agama Bab II	6.
7.	13 Mei 2022	ACC Seminar Proposal	7.
8.	29 Juli 2022	Konsultasi Revisi Seminar Proposal	8.
9.	13 Agustus 2022	Konsultasi Bab IV dan Script	9.
10.	30 September 2022	Konsultasi Revisi Bab IV	10.
11.	21 Oktober 2022	Konsultasi Bab IV dan V	11.
12.	25 November 2022	Konsultasi Revisi BAB IV dan V	12.
13.	02 Desember 2022	Konsultasi Bab I-V	13.
14.	06 Desember 2022	ACC Seminar Hasil	14.
15.	13 Desember 2022	Konsultasi Revisi Seminar Hasil	15.

KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

No	Tanggal	Hal	Tanda Tangan
16.	15 Desember 2022	Konsultasi Kajian Agama Bab IV	16.
17.	16 Desember 2022	Konsultasi Revisi Kajian Agama Bab IV	17.
18.	20 Desember 2022	Konsultasi Revisi Seminar Hasil dan Script	18.
19.	21 Desember 2022	ACC Sidang Skripsi	19.
20.	28 Desember 2022	ACC Skripsi untuk Syarat Yudisium	20.

Malang, 28 Desember 2022
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP. 19741129 200012 2 005