MODIFIKASI *AFFINE CIPHER* MENGGUNAKAN ALGORITMA BLUM-BLUM SHUB DALAM MENGAMANKAN PESAN

SKRIPSI

OLEH
SYAHLA AZNADILLAH
NIM. 18610054

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

MODIFIKASI *AFFINE CIPHER* MENGGUNAKAN ALGORITMA BLUM-BLUM SHUB DALAM MENGAMANKAN PESAN

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
Syahla Aznadillah
NIM. 18610054

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

MODIFIKASI *AFFINE CIPHER* MENGGUNAKAN ALGORITMA BLUM-BLUM SHUB DALAM MENGAMANKAN PESAN

SKRIPSI

Oleh
Syahla Aznadillah
NIM. 18610054

Telah Diperiksa dan Disetujui Untuk Diuji

Tanggal 16 Juni 2022

Dosen Pembimbing I	Dosen Pembimbing II
Evawati Alisah, M.Pd	Erna Herawati, M.Pd
NIP.19720604 199903 2 001	NIDT. 19760723 20180201 2 222

Mengetahui,
Ketua Program Studi Matematika

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

MODIFIKASI *AFFINE CIPHER* MENGGUNAKAN ALGORITMA BLUM-BLUM SHUB DALAM MENGAMANKAN PESAN

SKRIPSI

Oleh
Syahla Aznadillah
NIM. 18610054

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 21 Juni 2022

Ketua Penguji	: Prof. Dr. H. Turmudi, M.Si., Ph.D	.....
Anggota Penguji I	: Muhammad Khudzaifah, M.Si	.....
Anggota Penguji II	: Evawati Alisah, M.Pd	.....
Anggota Penguji III	: Erna Herawati, M.Pd	.....

Mengetahui,
Ketua Program Studi Matematika

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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan dibawah ini :

Nama	: Syahla Aznadillah
NIM	: 18610054
Program Studi	: Matematika
Fakultas	: Sains dan Teknologi
Judul Skripsi	: Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub dalam Mengamankan Pesan

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya 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 rujukan. Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 13 Juni 2022

Yang membuat pernyataan,

Syahla Aznadillah

NIM. 18610054

MOTO

“Pelangi yang muncul setelah hujan adalah janji alam bahwa masa buruk telah berlalu dan masa depan akan baik-baik saja.”

(Windry Ramadhina)

PERSEMBAHAN

Skripsi ini penulis persembahkan untuk :

Ayahanda Munadih dan Ibunda Sumiati tercinta yang senantiasa dengan ikhlas mendoakan, memberikan semangat, dan kasih sayang yang tak ternilai kepada penulis. Adik Muhamad Akhdan Najib dan Batrisya Zivana Hilyah yang telah memberikan dukungan kepada penulis.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Puji dan syukur kepada Allah Swt atas segala Rahmat-Nya sehingga penulis dapat menyelesaikan penyusunan draf skripsi yang berjudul "Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub dalam Mengamankan Pesan". Shalawat serta salam senantiasa tercurahkan kepada Nabi Muhammad Saw yang telah membimbing umatnya dari zaman jahiliah menuju zaman yang terang benderang.

Draf skripsi ini tidak akan terwujud tanpa orang-orang tercinta di sekeliling penulis yang mendukung dan membantu. Terima kasih dan doa terbaik penulis persembahkan kepada:

1. Prof. Dr. H. M. Zainuddin, M.A., selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim.
2. Dr. Sri Harini, M.Si, selaku dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim.
3. Dr. Elly Susanti, S.Pd., M.Sc, selaku ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim.
4. Evawati Alisah, M.Pd, selaku dosen pembimbing I yang senantiasa mengayomi dan membimbing penulis.
5. Erna Herawati, M.Pd, selaku dosen pembimbing II yang telah memberikan arahan dan ilmunya kepada penulis.
6. Prof. Dr. H. Turmudi, M.Si. Ph.D, selaku ketua penguji yang telah memberikan kritik, saran serta dukungannya kepada penulis.
7. Muhammad Khudzaifah, M.Si, selaku ketua penguji yang telah memberikan kritik, saran serta dukungannya kepada penulis.
8. Seluruh dosen dan staf administrasi Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim.
9. Ayahanda tercinta Munadih, Ibunda tercinta Sumiati, Adik tercinta Muhamad Akhdan Najib dan Batrisya Zivana Hilyah serta seluruh keluarga tercinta yang selalu memberikan do'a, semangat serta motivasi kepada penulis.

10. Teman tercinta Aulia Nanda, Nabilah Izaturizqi, Hazimah, dan Afidatul Masbakhah yang selalu menemani, membantu dan memberikan dukungan kepada penulis sampai saat ini.
11. Seluruh teman-teman KKN Pegadungan, tim PKL LKPP, dan Matematika 2018 yang telah memberikan semangat serta bantuan kepada penulis.
12. Seluruh pihak yang tidak dapat disebutkan satu-persatu, yang telah membantu penulis dalam menyelesaikan skripsi ini.

Penulis menyadari bahwa penelitian ini masih jauh dari kata sempurna, dikarenakan oleh keterbatasan, kemampuan dan pengetahuan penulis. Semoga Allah SWT melimpahkan rahmat-Nya kepada kita semua agar draf skripsi ini dapat bermanfaat bagi peneliti serta pembaca. Aamiin.

Wassalamualaikum Warahmatullahi Wabarakatuh

Malang, ...

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR SIMBOL

X_i = Bilangan Acak
z_i = Bilangan Acak Baru
m = Kunci Multiplikatif
m^{-1} = Balikan Modulo Kunci Multiplikatif
y = Jumlah Indeks Karakter Tabel ASCII
b = Nilai Pergeseran Aditif
P = Indeks Karakter *Plaintext*
C = Indeks Karakter *Ciphertext*

DAFTAR LAMPIRAN

ABSTRAK

Aznadillah, Syahla. 2022. **Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub Dalam Mengamankan Pesan**. Skripsi Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing (1) : Evawati Alisah, M.Pd, Pembimbing (2) : Erna Herawati, M.Pd.

Kata Kunci : Modifikasi, *Affine Cipher*, Algoritma, Blum-Blum Shub.

Kriptografi merupakan ilmu menulis kode secara rahasia menggunakan model matematika tertentu. Fungsi utama kriptografi yaitu untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain. Keamanan pesan menjadi syarat penting dalam bertukar informasi. Hal ini dikarenakan banyaknya kejahatan internet sehingga para pengguna merasa tidak aman setiap kali mengirimkan pesan. Oleh karena itu, diperlukan solusi yang bisa membantu agar pesan atau informasi yang diperlukan bersifat aman sampai ke tujuan sesuai dengan yang diinginkan. Penelitian ini membahas tentang modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub dalam mengamankan pesan. Proses modifikasi dilakukan dengan cara men-*generate* bilangan acak yang dihasilkan algoritma Blum-Blum Shub dan bilangan acak tersebut akan digunakan untuk menggantikan nilai pergeseran aditif pada proses enkripsi dan dekripsi *Affine Cipher*. Bilangan acak yang dihasilkan algoritma Blum-Blum Shub sebanyak jumlah posisi huruf pada *plaintext*. Sehingga kunci enkripsi yang digunakan setiap hurufnya akan berbeda-beda dan tidak terjadi perulangan. Maka dari itu, *ciphertext* yang dihasilkan dari proses modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub lebih rumit dan lebih susah untuk dipecahkan oleh kriptanalis.

ABSTRACT

Aznadillah, Syahla, 2022. **Affine Cipher Modification uses The Blum-Blum Shub Algorithm to Secure a Message**. Thesis. Mathematics Study Program. Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Supervisor: (I) Evawati Alisah, M.Pd., (II) Erna Herawati, M.Pd.

Keywords: Modification, Affine Cipher, Algorithm, Blum-Blum Shub.

Cryptography is the science of writing code in secret using specific mathematical models. The primary function of cryptography is to maintain the security of messages when messages are sent from one place to another. Message security is an essential requirement in exchanging information. This is because of so many internet crimes that users feel unsafe whenever they send a message. Therefore, a solution is needed to help ensure the message or information conveyed is safe and reaches the desired destination. This study discusses the Affine Cipher modification using the Blum-Blum Shub algorithm in securing messages. The modification process is carried out by generating random numbers generated by the Blum-Blum Shub algorithm, and these random numbers will be used to replace the additive shift value in the Affine Cipher encryption and decryption process. The random number generated by the Blum-Blum Shub algorithm is as much as the number of letter positions in the plaintext. So that the encryption key used for each letter will be different, and there will be no repetition. Therefore, the ciphertext generated from the Affine Cipher modification process using the Blum-Blum Shub algorithm is more complicated for cryptanalysts to solve.

ملخص

أزنادة الله، سياهلة، 2022. يستخدم تعديل Affine Cipher خوارزمية Blum-Blum Shub لتأمين رسالة. أطروحة. برنامج دراسة الرياضيات. كلية العلوم والتكنولوجيا، جامعة الإسلام نيغيري مولانا مالك إبراهيم مالانغ. المشرف: (الأول) إيفاواتي أليسا، دكتوراه في الطب، (الثاني) إرنا هيراواتي، دكتوراه في الطب.

الكلمات الرئيسية: تعديل، Affine Cipher، Algoritma، Blum-Blum Shub.

علم التشفير هو علم كتابة الشفرة في السر باستخدام نماذج رياضية محددة. تتمثل الوظيفة الأساسية للتشفير في الحفاظ على أمان الرسائل عند إرسال الرسائل من مكان إلى آخر. وأمن الرسائل شرط أساسي في تبادل المعلومات. هذا بسبب العديد من جرائم الإنترنت التي يشعر المستخدمون بعدم الأمان كلما أرسلوا رسالة. لذلك، هناك حاجة إلى حل للمساعدة في ضمان سلامة الرسالة أو المعلومات المنقولة والوصول إلى الوجهة المرغوبة. تناقش هذه الدراسة تعديل Affine Cipher باستخدام خوارزمية Blum-Blum Shub في تأمين الرسائل. تتم عملية التعديل عن طريق توليد أرقام عشوائية تم إنشاؤها بواسطة خوارزمية Blum-Blum Shub، وسيتم استخدام هذه الأرقام العشوائية لاستبدال قيمة التحول المضافة في عملية تشفير Affine Cipher وفك التشفير. الرقم العشوائي الذي تم إنشاؤه بواسطة خوارزمية Blum-Blum Shub هو بقدر عدد مواقع الحروف في النص العادي. بحيث يكون مفتاح التشفير المستخدم لكل حرف مختلفًا، ولن يكون هناك تكرار. لذلك، فإن النص المشفر الناتج عن عملية تعديل Affine Cipher باستخدام خوارزمية Blum-Blum Shub أكثر تعقيدًا لحل التحاليل المشفرة.

BAB I
PENDAHULUAN

1.1 Latar Belakang

Kriptografi merupakan teknik menyandikan pesan yang dapat terbaca dan dipahami (*Plaintext*) menjadi pesan rahasia yang tidak dapat dibaca atau dipahami (*Ciphertext*) disebut dengan teknik enkripsi. Sebaliknya, kriptografi juga dapat mengubah pesan rahasia yang tidak dapat dipahami (*Ciphertext*) menjadi pesan asli yang dapat dipahami (*Plaintext*) atau dinamakan dengan teknik dekripsi. Teknik enkripsi dan dekripsi tersebut menggunakan model matematika tertentu seperti kunci (Cipher), algoritma, dan generator. Kriptografi terdiri dari 3 macam yaitu kriptografi simetris, asimetris dan *hybrid*. Kriptografi simetris adalah algoritma yang menggunakan kunci pada proses enkripsi sama dengan kunci pada proses dekripsi. Kriptografi simetris atau yang disebut kriptografi klasik ini mengilustrasikan bahwa pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan, sehingga penerima pesan bisa mendekripsi pesan tersebut. Sebagai contoh *Caesar Cipher, Affine Cipher*, *Vigenere Cipher,* dan lainnya. Kebalikan dari kriptografi simetris, kriptografi asimetris atau yang dikenal dengan kriptografi modern adalah kriptografi yang memiliki kunci enkripsi yang digunakan tidak sama dengan kunci dekripsi. Contoh kriptografi asimetris antara lain RSA, Elgamal, Elliptic Curve dan sebagainya. Sedangkan, gabungan dari kedua kriptografi disebut dengan kriptografi *hybrid* (Munir, 2019).

Fungsi utama ilmu kriptografi yaitu untuk memberikan keamanan pada suatu sistem dalam mengirimkan pesan agar pesan tersebut tidak dapat diretas oleh kriptanalis. Keamanan pesan atau informasi menjadi syarat yang harus dipenuhi oleh semua pihak yang terlibat dalam sistem tersebut.

Kemajuan teknologi juga memberikan kemudahan kepada setiap pihak dalam bertukar pesan atau informasi. Kebutuhan akan informasi dan komunikasi merupakan hal yang tidak kalah pentingnya dari kebutuhan sandang dan pangan manusia meskipun, peranan informasi dalam beberapa dekade kurang mendapat perhatian. Pertukaran informasi tersebut juga memberikan dampak positif dan negatif. Dampak positif dari kemajuan teknologi dalam bertukar pesan atau informasi ini seperti penyebaran informasi yang sangat luas dan cepat dari berbagai bidang memberikan perubahan yang amat cepat dalam kehidupan manusia. Hal ini juga menimbulkan dampak negatif yang bisa terjadi dikarenakan kurangnya keamanan yang digunakan dan mencegah *cryptanalysis*. Dengan adanya kejahatan-kejahatan internet ini para pengguna merasa semakin tidak aman setiap kali mengirimkan pesan. Maka diperlukan solusi yang bisa membantu agar pesan atau informasi yang dipertukarkan bersifat aman dan sampai ketujuan sesuai dengan yang diinginkan. Sebagaimana sesuai dengan konsep amanah bahwa merahasiakan pesan seseorang itu termasuk menjaga amanah dimana hanya penerima pesan tersebut lah yang boleh mengetahuinya. Dalam surah "Al Mu'minun" ayat 8 yang artinya:

"*Dan orang-orang yang memelihara amanah-amanah (yang dipikulnya) dan janjinya*"

Ayat tersebut menerangkan bahwasanya seseorang dapat dikatakan orang mukmin jika orang tersebut suka memelihara amanah-amanah dari Allah dan juga sesama manusia. Bilamana seseorang dititipkan sesuatu baik itu barang penting atau tidak, uang dan lain sebagainya sebagai amanah yang harus disampaikan kepada orang lain, maka orang tersebut harus benar-benar menyampaikan amanah

itu sebagaimana mestinya. Oleh karena itu, orang mukmin tersebut sungguh beruntung semasa hidupnya hingga di akhirat kelak.

Penelitian yang dilakukan oleh Borodzhieva dan Manoilov (2014) membahas bahwa *Affine Cipher* memiliki kelemahan yang sama dengan *Caesar Cipher* yaitu ukuran kunci yang kecil. Ukuran ruang kunci algoritma *Affine* adalah $12 \times 26 = 312$ (12 adalah jumlah bilangan bulat yang memiliki *invers* di $\mathbb{Z}_{26}$ karena $(a, 26) = 1$, sedangkan untuk kunci b terdapat 26 nilai kemungkinan). Algoritma *Affine* dapat dipecahkan dengan metode pencarian *brute force* karena ukuran ruang kuncinya yang kecil. Sedangkan, penelitian lain yang dilakukan oleh Ofelius Laia (2019) membahas tentang proses enkripsi pesan teks menggunakan *Affine Cipher* yang dimodifikasikan dengan algoritma *Linear Congruent Generator* (*LCG*) dimana setiap *plaintext* dikonversikan ke dalam bentuk kode ASCII dan masing-masing karakter ditempatkan ke dalam kolom ganjil sedangkan kolom genap diisi dengan bilangan yang dihasilkan oleh *LCG*. Selanjutnya membangkitkan kembali bilangan acak yang dihasilkan dari bahasa pemrograman. Kemudian lakukan enkripsi menggunakan *Affine Cipher* dan angka yang dihasilkan di konversikan kembali ke dalam indeks karakter ASCII. Jumlah huruf pada *ciphertext* ini menghasilkan dua kali lebih banyak dari jumlah huruf *plaintext*-nya. Secara teoritis, *LCG* dapat menghasilkan bilangan acak yang lumayan baik, namun masih perlu berhati-hati dalam pemilihan nilai a, b, dan m. Pemilihan nilai-nilai tersebut berakibat buruk karena dapat menghasilkan bilangan acak yang mudah diprediksi. Oleh karena itu, *LCG* kurang aman digunakan pada bidang kriptografi.

Metode kriptografi yang menjadi objek penelitian ini adalah kriptografi simetris yaitu *Affine Cipher*. *Affine Cipher* dipilih karena proses enkripsi dan dekripsi yang dihasilkan lebih cepat dibandingkan dengan kriptografi asimetris yang mana *Affine Cipher* memiliki 2 kunci yang berbeda yaitu kunci multiplikatif dan aditif. Namun, kelemahan *Affine Cipher* adalah kunci yang mudah digunakan dan mudah ditebak. Maka dari itu, peneliti ingin memodifikasi *Affine Cipher* dengan algoritma Blum-Blum Shub. Salah satu modifikasi yang dilakukan yakni men-*generate* bilangan acak yang dihasilkan oleh algoritma Blum-Blum Shub dan nantinya bilangan acak tersebut akan digunakan untuk menggantikan nilai pergeseran aditif pada proses enkripsi dan dekripsi *Affine Cipher*. Algoritma Blum-Blum Shub dipilih karena keamanan Blum-Blum Shub terletak pada sulitnya memfaktorkan *n* dan menghasilkan bilangan yang tidak mudah diprediksi.

Berdasarkan uraian yang telah dikemukakan, peneliti tertarik mengkaji lebih dalam penelitian yang berjudul "Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub dalam Mengamankan Pesan".

1.2 Rumusan Masalah

Berdasarkan latar belakang, rumusan masalah penelitian ini sebagai berikut:

1. Bagaimana proses modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub ?
2. Bagaimana proses enkripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub dalam mengamankan pesan ?

3. Bagaimana proses dekripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub dalam mengamankan pesan ?

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah, tujuan penelitian ini sebagai berikut :

1. Untuk mengetahui proses modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub.
2. Untuk mengetahui proses enkripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub dalam mengamankan pesan.
3. Untuk mengetahui proses dekripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub dalam mengamankan pesan.

1.4 Manfaat Penelitian

Berdasarkan tujuan penelitian, peneliti dapat memberikan manfaat kepada siapapun khususnya pembaca dan penulis antara lain :

1. Bagi penulis

   Mengetahui cara mengamankan pesan menggunakan *Affine Cipher* yang dimodifikasi oleh algoritma Blum-Blum Shub.
2. Bagi pembaca dan peneliti selanjutnya
   a. Dapat menambah wawasan tentang ilmu kriptografi khususnya *Affine Cipher* yang dimodifikasi oleh algoritma Blum-Blum Shub.
   b. Mengetahui keamanan pesan dengan menggunakan *Affine Cipher* yang dimodifikasi oleh algoritma Blum-Blum Shub.

c. Sebagai referensi bagi peneliti selanjutnya dalam memodifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub.

3. Bagi Institusi

a. Sebagai media pembelajaran bagi para mahasiswa khususnya mata kuliah Kriptografi.

b. Mengimplementasikan materi khususnya mata kuliah Kriptografi dalam dunia teknologi.

1.5 Batasan Masalah

Adapun batasan masalah penelitian ini sebagai berikut :

1. Mengenkripsi dan mendekripsi huruf, angka, dan simbol pada tabel ASCII.
2. Pengurutan posisi suatu pesan dimulai dari 1 hingga jumlah huruf pada pesan.
3. Pengurutan indeks karakter pesan dimulai dari 0 hingga 255.
4. Bilangan prima yang digunakan yaitu $p = 6427$, $q = 8999$ dan $s = 2221$.
5. Hasil enkripsi dan dekripsi berupa huruf, angka, dan simbol pada tabel ASCII.

1.6 Definisi Istilah

Beberapa istilah yang digunakan dalam penelitian ini adalah :

1. Pesan adalah informasi berupa teks.
2. *Plaintext* adalah teks jelas atau pesan asli yang dapat dipahami.
3. *Ciphertext* adalah suatu pesan yang telah melalui proses enkripsi dan tidak memiliki arti (makna).

4. Enkripsi adalah proses suatu pesan asli (*plaintext*) yang diubah dengan algoritma tertentu sehingga menjadi kode rahasia yang tidak dimengerti (*ciphertext*).
5. Dekripsi adalah pesan atau kode rahasia yang tidak dapat terbaca (*ciphertext*) diubah menjadi pesan yang dapat dibaca (*plaintext*) dengan menggunakan algoritma tertentu.
6. Cipher adalah fungsi matematika yang digunakan untuk enkripsi dan dekripsi.
7. Kunci adalah parameter yang digunakan di dalam enkripsi dan dekripsi.
8. Algoritma adalah proses atau serangkaian aturan yang harus diikuti dalam perhitungan atau operasi pemecahan masalah.
9. Indeks adalah penomoran suatu huruf, angka, dan simbol sesuai dengan tabel ASCII.
10. Bit adalah satuan terkecil dalam sistem angka biner.
11. Generator adalah suatu pembangkit kunci yang akan digunakan dalam proses enkripsi dan dekripsi.
12. *Cryptanalysis* adalah ilmu mempelajari teknik matematika untuk memecahkan pesan rahasia tanpa memiliki kunci.

BAB II
KAJIAN PUSTAKA

2.1 Teori Pendukung

2.1.1 Keterbagian

Keterbagian merupakan bagian dasar dari berbagai sifat teori bilangan. Kajian sifat-sifat dari teori bilangan tersebut merupakan dasar pengembangan teori bilangan yang berkaitan dengan keterbagian (*divisibility*). Bilangan bulat yang dibagi oleh bukan bilangan bulat menghasilkan suatu bilangan bulat lain atau bilangan yang bukan termasuk bilangan bulat.

Definisi 2.1

Misalkan dua bilangan $a, b \in \mathbb{Z}$ dengan $a \neq 0$, maka a disebut membagi b dinotasikan $a|b$ apabila $b = ak$, untuk suatu $k \in \mathbb{Z}$ (Irawan, 2014).

Contoh :

1. $2|6$, sebab ada $3 \in \mathbb{Z}$, sehingga $6 = 2 \times 3$
2. $8|40$, sebab ada $5 \in \mathbb{Z}$, sehingga $40 = 8 \times 5$

Berdasarkan definisi 2.1, bilangan bulat a dengan $a \neq 0$ dapat membagi bilangan bulat b dinotasikan $a|b$ artinya "a membagi b" atau "b habis dibagi a" atau "a pembagi b" atau "a faktor dari b" atau "b kelipatan dari a". Jika a tidak dapat membagi b, dinotasikan $a \nmid b$. Jika $a|b$ dan $0 < a < b$, maka a disebut pembagi sejati dari b.

Teorema 2.1 :

Diberikan $k, l, m \in \mathbb{Z}$.

1. Jika $k|l$ maka $k|lx$ untuk setiap $x \in \mathbb{Z}$; $k \neq 0$
2. Jika $k|l$ dan $l|m$, maka $k|m$; $k \neq 0$, $l \neq 0$

3. Jika $k|l$ dan $k|m$, maka $k|l \pm m$; $k \neq 0$
4. Jika $k|l$ dan $k|m$, maka $k|(lx \pm my)$ untuk setiap $x, y \in \mathbb{Z}$; dengan $k \neq 0$.
5. Jika $k|l$ dan $l|k$, maka $k = \pm l$, $k \neq 0$, $l \neq 0$
6. Jika $k|l, k > 0$, dan $l > 0$, maka $k \leq l$; $k \neq 0$
7. Untuk suatu $a \in \mathbb{Z}$ dan $a \neq 0, k|l$ jika dan hanya jika $ak|al$ dengan $a \neq 0$

2.1.2 Aritmatika Modulo

Aritmatika modulo dapat didefinisikan sebagai operasi aritmatika yang memetakan semua bilangan bulat ke dalam himpunan bilangan bulat dalam batas-batas himpunan. Misalkan $a \in \mathbb{Z}$ dan $n \in \mathbb{Z}$ dan $n > 0$. Operasi $a\ (mod\ n)$ (dibaca "$a\ modulo\ n$") memberikan sisa apabila a dibagi dengan m. Bilangan m disebut modulo dan hasil operasi modulo m terletak dalam himpunan $\{0, 1, 2, \ldots, n-1\}$(Stallings, 2003).

Notasi : $a\ (mod\ n) \equiv r$ sedemikian sehingga $a = nq + r$, dengan $0 \leq r < n$.

Berikut sifat-sifat aritmatika modulo :

1. $[a\ (mod\ n) + b(mod\ n)](mod\ n) = (a + b)(mod\ n)$
2. $[a(mod\ n) - b(mod\ n)](mod\ n) = (a - b)(mod\ n)$
3. $[a(mod\ n) \times b(mod\ n)](mod\ n) = (a \times b)(mod\ n)$

Aturan aritmatika biasa yang melibatkan penambahan, pengurangan, dan perkalian termasuk ke dalam aritmatika modulo.

Tabel 2.1 Sifat-sifat Aritmatika Modulo

Sifat-Sifat	Pernyataan
Hukum komutatif	$(k + x)(mod\ n) = (x + k)(mod\ n)$ $(k \times x)(mod\ n) = (x \times k)(mod\ n)$
Hukum asosiatif	$[(k + x) + y](mod\ n) = [k + (x + y)](mod\ n)$ $[(k \times x) \times y](mod\ n) = [k \times (x \times y)](mod\ n)$
Hukum distributif	$[k \times (x + y)](mod\ n) = [(k \times x) + (k \times y)](mod\ n)$ $[k + (x \times y)](mod\ n) = [(k + x) \times (k + y)](mod\ n)$
Identitas	$(0 + k)(mod\ n) = k\ (mod\ n)$ $(1 \times k)(mod\ n) = k\ (mod\ n)$
Invers aditif$(-k)$	Untuk setiap $k \in Z_n$, ada l bahwa $k + l \equiv 0\ (mod\ n)$

2.1.3 Pembagi Bersama Terbesar (PBB)

Definisi 2.3

Pembagi bilangan bulat positif terbesar dapat membagi kedua bilangan bulat a dan b yang disebut dengan pembagi bersama terbesar (PBB) atau *greatest common divisor* (*gcd*) dari a dan b (Batten, 2012).

Bilangan bulat b dan c (keduanya tidak nol) maka bilangan bulat a disebut pembagi bersama terbesar dari b dan c, jika dipenuhi (Irawan, 2014):

1. $a > 0$
2. $a|b$ dan $a|c$
3. Jika $d|b$ dan $d|c$ maka $d|a$

Notasi : Pembagi bersama terbesar dari b dan c dituliskan $a = (b, c)$ dan karena $a > 0$ maka $a = (b, c) \geq 1$ sehingga $(b, c) = (b, -c) = (-b, c) = (-b, -c)$.

Contoh :

Pembagi bersama terbesar dari 18 dan 12 adalah 6 ditulis $(18,12) = 6$

Begitu pula $(18, -12) = (-18,12) = (-18,12) = (-18, -12) = 6$.

2.1.4 Kongruensi

Kongruensi dapat diartikan sebagai salah satu gagasan pembagian bilangan bulat dalam teori bilangan dengan menggunakan konsep kongruensi yang ada.

Definisi 2.4

Misalkan $m \in \mathbb{N}$ dan $a, b \in \mathbb{Z}$ dikatakan kongruen jika m membagi $(a - b)$ dapat ditulis : $a \equiv b \ (mod\ m)$. Dalam kasus ini, m dikatakan modulo kongruensi dan a kongruen dengan b (Das, 2013).

Contoh :

1. $27 \equiv 2 \ (mod\ 5)$ karena $(27 - 2)$ dapat dibagi oleh 5
2. $49 \not\equiv 3 \ (mod\ 7)$ karena $(49 - 3)$ tidak dapat dibagi oleh 7

Berdasarkan definisi dan contoh diatas, relasi kongruensi mempunyai kemiripan sifat dengan persamaan, sebab relasi kongruensi dapat dinyatakan sebagai persamaan. Oleh karena itu terdapat proporsi dalam kongruensi.

Proporsi 2.1

Misalkan $m \in \mathbb{N}$ dan $x, y, z \in \mathbb{Z}$ adalah sembarang

a. $x \equiv x \ (mod\ m)$
b. Jika $x \equiv y \ (mod\ m)$ maka $y \equiv x \ (mod\ m)$
c. Jika $x \equiv y \ (mod\ m)$ dan $y \equiv z \ (mod\ m)$ maka $z \equiv x \ (mod\ m)$
d. Jika $x \equiv z \ (mod\ m)$ dan $y \equiv p \ (mod\ m)$ maka $x + y \equiv (z + p)(mod\ m)$, $x - y \equiv (z - p) \ (mod\ m)$, dan $xy \equiv zp \ (mod\ m)$
e. Jika $x \equiv y \ (mod\ m)$ dan $f(a)$ adalah polinomial dengan koefisien bilangan bulat, maka $f(x) \equiv f(y)(mod\ m)$
f. Jika $x \equiv y \ mod\ m$ dan a adalah bilangan bulat positif dari m, maka $x \equiv y \ (mod\ a)$

g. $xy \equiv xz\ (mod\ m)$ jika hanya jika $y \equiv z\ \left(mod\ \frac{m}{(x,m)}\right)$

2.1.5 Balikan Modulo

Suatu bilangan bulat $a\ (mod\ m)$ memiliki balikan (*invers*) jika hanya jika $(a, m) = 1$ dan $m > 1$. Balikan dari $a\ (mod\ m)$ adalah sebuah bilangan bulat a^{-1} sedemikian sehingga $a \times a^{-1} \equiv 1\ (mod\ m)$ (Munir, 2019). Mencari suatu balikan (*invers*) dari $a\ (mod\ m)$, harus memuat algoritma Euclidean dari a dan m sama dengan 1. Koefisien p merupakan balikan dari $a\ (mod\ m)$, semua bilangan bulat yang kongruen dengan $p\ (mod\ m)$ juga merupakan balikan modulo dari a (Irawan, 2014).

Contoh :

Mencari balikan dari 3 $(mod\ 11)$.

Karena $(11, 5) = 1$, maka terdapat $5^{-1}\ (mod\ 11)$. Berdasarkan algoritma Euclidean bahwa $11 = 2 \times 5 + 1$. Setelah itu, bentuk persamaan tersebut menjadi

$$(-2 \times 5) + (1 \times 11) = 1$$

Maka diperoleh balikan dari $5\ (mod\ 11)$ yaitu -2 atau $5^{-1}\ (mod\ 11) \equiv (-2)(mod\ 11)$.

2.1.6 Algoritma Euclid

Algoritma Euclid termasuk salah satu algoritma tertua dan sangat berguna di semua teori bilangan. Algoritma ini dapat memfaktorkan angka dengan jumlah yang besar. Misalkan a dan b adalah bilangan bulat tidak negatif dan asumsikan bahwa $b \neq 0$. Maka dapat peneliti hitung :

$$a = q_1 b + r_1, dimana\ 0 \leq r_1 \leq b$$
$$b = q_2 r_1 + r_2, dimana\ 0 \leq r_2 \leq r_1$$
$$r_1 = q_3 r_2 + r_3, dimana\ 0 \leq r_3 \leq r_2$$
$$\vdots$$
$$r_{n-3} = q_{n-1} r_{n-2} + r_{n-1}, dimana\ 0 \leq r_{n-1} \leq r_{n-2}$$
$$r_{n-2} = q_n r_{n-1} + 0.$$

Sisa pembagi terakhir yang bukan nol dinamakan r_{n-1}, atau sama dengan $(a, b) = 1$ (Kraft & Washington, 2015).

Contoh :

Misalkan $a = 12345$, $b = 11111$ maka dapat menghitung *invers modulo* menggunakan algoritma Euclidean sebagai berikut:

$$\begin{aligned} 12345 &= 11111 \times 1 + 1234 \\ 11111 &= 1234 \times 9 + 5 \\ 1234 &= 5 \times 246 + 4 \\ 5 &= 4 \times 1 + 1 \\ 4 &= 1 \times 4 \end{aligned}$$

Karena $(12345,11111) = 1$ peneliti menulis 1 sebagai sebuah kombinasi linier dari 12345 dan 11111 maka :

$$\begin{aligned} 1 &= 5 - 4 \times 1 \\ &= 5 - [1234 - 5 \times 246] \times 1 \\ &= -1234 + 5 \times 247 \\ &= -1234 + [11111 - 1234 \times 9] \times 247 \\ &= 11111 \times 247 - 1234 \times 2224 \\ &= 11111 \times 247 - [12345 - 11111 \times 1] \times 2224 \\ &= -1234 \times 2224 + 11111 \times \mathbf{2471} \end{aligned}$$

Didapat *invers modulo* dari $11111\ (mod\ 12345)$ yaitu 2471.

2.1.7 Kriptografi

2.1.7.1 Definisi Kriptografi

Kriptografi merupakan seni dan ilmu mengubah data atau teks biasa menjadi urutan bit yang muncul sebagai acak. Ilmu kriptografi dapat didefinisikan pula sebagai ilmu menulis kode secara rahasia dan merupakan seni kuno. Dalam bahasa Yunani, kriptografi terbagi menjadi dua yaitu *kripto* yang berarti rahasia dan *graphia* yang berarti tulisan. Menurut terminologi kriptografi adalah ilmu dan seni untuk menjaga keamanan pesan ketika pesan dikirim dari suatu tempat ke tempat lain (Jamaludin & Romindo, 2020).

Kata "*cryptography*", "*cryptology*", dan "*cryptanalysis*" memiliki arti yang berbeda. Dimulai dengan kata "*crypt*" yang berarti tersembunyi dan berakhir dengan kata "*graphy*" yang berarti tulisan. Jadi jika digabungkan memiliki arti "tulisan yang tersembunyi. Biasanya mengarah untuk enkripsi yang membangun sistem untuk transmisi yang aman. Kata yang dienkripsikan kita sebut "Cipher" atau "*Ciphertext*" (Batten, 2012).

Kata *cryptanalysis* mengacu pada analisis tentang hal-hal yang tersembunyi atau cipher untuk mengungkapkan apa yang tersembunyi. Kata ini biasanya mengacu pada komponen dekripsi atau penemuan sistem ketika analis tidak memiliki kunci yang sah untuk membaca sandi (Batten, 2012).

Kata *cryptology* memuat 2 komponen yaitu "tersembunyi" dan "belajar" yang mengacu pada ilmu tentang tulisan tersembunyi atau rahasia. Pada dasarnya, kata *cryptology* ini akan menjadi kata yang tepat untuk berdiskusi tentang teknik enkripsi dan analisisnya. Tetapi kebanyakan orang menggunakan kata *cryptography* (Batten, 2012).

2.1.7.2 Klasifikasi Kriptografi

Berdasarkan kunci yang digunakan, kriptografi terdiri dari dua jenis yaitu, kriptografi simetris (*symmetric-key cryptography*) dan kriptografi asimetris (*asymmetric-key cryptography*). Kunci yang digunakan pada kriptografi simetri merupakan kunci privat (*private-key cryptography*) atau kunci rahasia (*secret-key cryptography*). Kriptografi asimetris juga disebut kriptografi konvensional (*conventional cryptography*). Pendapat lain berpendapat bahwa pengirim dan penerima pesan sudah berbagi kunci yang sama sebelum bertukar pesan, sehingga penerima sudah mengetahui bagaimana cara membaca kunci tersebut. Dengan kata lain, kriptografi kunci simetri ini yaitu kunci yang dipakai pada proses enkripsi sama dengan kunci yang dipakai pada proses dekripsi. Dalam kriptografi simetri, keamanannya bergantung pada kerahasiaan kuncinya, sedangkan algoritmanya sendiri tidak perlu rahasia. Contoh dari kunci simetri antara lain, *Caesar Cipher*, Cipher transposisi, *Vigenere Cipher*, *Playfair Cipher*, *Affine Cipher*, *Hill Cipher*, *One-Time Pad*, dan lain sebagainya (Munir, 2019).

Kriptografi asimetri (*asymmetric-key cryptography*) atau yang disebut dengan kriptografi kunci publik (*public-key cryptography*) merupakan kunci yang digunakan pada proses enkripsi tidak sama dengan kunci yang digunakan pada proses dekripsi. Kunci untuk enkripsi dapat diketahui oleh siapapun, sedangkan kunci dekripsi harus dirahasiakan. Pada kriptografi kunci publik, setiap pihak yang berkomunikasi memiliki sepasang kunci, yaitu kunci privat dan kunci publik. Pengirim mengenkripsi pesannya dengan menggunakan kunci publik si penerima pesan. Hanya penerima pesan yang dapat mendekripsi

ciphertext karena hanya penerima yang mengetahui kunci privatnya sendiri. Keuntungan kriptografi kunci publik yaitu tidak ada kebutuhan untuk mendistribusikan kunci rahasia (kunci privat) serta jumlah kunci dapat ditekan. Contoh algoritma kriptografi asimetri diantaranya *RSA, Elgamal, DSA, Diffie-Helman, Elliptic Curve Cryptography*, dan sebagainya (Munir, 2019).

2.1.8 Algoritma Kriptografi

Kata algoritma muncul dalam kamus Webster pada akhir tahun 1957 yang mempunyai arti proses perhitungan dengan bahasa Arab. Algoritma berasal dari kata algorism, ditemukan oleh salah satu penulis Arab yang terkenal yaitu Abu Ja'far Muhammad ibnu Musa al-Khuwarizmi (al-Khuwarizmi). Kata algorism lambat laun berubah menjadi algorithm.

Definisi terminologi Algoritma merupakan urutan langkah-langkah logis untuk menyelesaikan masalah yang disusun secara sistematis. Sedangkan algoritma kriptografi dapat diartikan sebagai langkah atau proses bagaimana merahasiakan pesan dari orang-orang yang tidak berhak membacanya. Algoritma kriptografi terdiri dari tiga fungsi dasar yaitu (Ariyus, 2006):

- **Enkripsi** adalah salah satu fungsi dasar yang sangat penting dalam kriptografi dimana fungsinya untuk mengamankan data yang akan dikirim dan harus terjaga kerahasiannya. Pesan asli (*plaintext*) yang diubah menjadi pesan atau kode-kode yang tidak dimengerti. Proses enkripsi memerlukan suatu cipher atau kode yang nantinya digunakan untuk menghitung kunci *ciphertext*. Seperti halnya dengan kita yang tidak mengerti akan sebuah kata asing, maka kita akan melihatnya di kamus atau daftar istilah-istilah. Berbeda dengan enkripsi, untuk merubah suatu

plaintext ke bentuk *ciphertext* kita menggunakan algoritma yang dapat mengkodekan pesan yang akan kita rahasiakan.

- **Dekripsi** adalah suatu pesan atau kode rahasia yang telah dienkripsi dikembalikan ke bentuk asalnya yaitu pesan asli (*Plaintext*). Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi.
- **Kunci** adalah sebuah kode yang dipakai dalam melakukan enkripsi dan dekripsi, kunci terbagi menjadi dua jenis yaitu kunci pribadi (*private key*) dan kunci umum (*publik key*).

2.1.9 *Affine Cipher*

Affine Cipher termasuk dalam *monoalphabetic substitution cipher* dimana setiap huruf dalam alfabet dikonversikan ke dalam angka-angka yang disebut dengan indeks karakter kemudian dienkripsi secara sederhana dengan suatu persamaan dan dikonversi kembali ke huruf (Kromodimoeljo, 2010). Kunci pada *Affine Cipher* menggunakan dua bilangan bulat m dan b, dimana nilai m yang dapat dipakai adalah anggota elemen pada $\mathbb{Z}_{26}$ yang memiliki *invers* memenuhi $(m, 26) = 1$ (Sadikin, 2012).

Affine Cipher merupakan perluasan dari *Caesar Cipher*, melakukan proses enkripsi dengan cara mengalikan *plaintext* yang disimbolkan P dengan sebuah nilai m $((m, y) = 1)$. Selanjutnya menambahkan hasil yang telah diperoleh dengan sebuah nilai acak b. Nilai m disebut nilai pergeseran multiplikatif, sedangkan b disebut nilai pergeseran aditif. Secara matematis enkripsi *plaintext P* yang menghasilkan *ciphertext* C dinyatakan dengan persamaan :

$$C = (mP + b)(mod\ y)$$

Contoh :

Misalkan *plaintext* ALJABAR yang akan dikirim dan perlu dienkripsi *plaintext* tersebut dengan mengambil $m = 5$ $((26,5) = 1)$ dan $b = 9$. Indeks karakter yang digunakan yaitu dimulai dari $\mathrm{A} = 0$ hingga $\mathrm{Z} = 26$. Perhitungan enkripsi adalah sebagai berikut:

Untuk huruf A = 0 maka kita enkripsikan

$$5 \times 0 + 9 = 9 \equiv 9 \ (mod\ 26)$$

menjadi A→J

Tabel 2.2 Contoh Enkripsi *Affine Cipher*

Plaintext	A	L	J	A	B	A	R
Indeks Karakter	0	11	9	0	1	0	17
$(5P + 9)(mod\ 26)$	9	12	2	9	14	9	16
Ciphertext	J	M	C	J	O	J	Q

Dalam hal ini, n merupakan ukuran alfabet yang digunakan dan m adalah bilangan bulat dengan syarat $(m, n) = 1$ sedangkan b adalah jumlah pergeseran. Untuk melakukan dekripsi, persamaan harus dipecahkan untuk memperoleh nilai P. Solusinya, hanya ada jika balikan atau *invers* dari $m\ (mod\ y)$ dinotasikan dengan m^{-1}. Dekripsi *Affine Cipher* dapat dilakukan dengan persamaan berikut (Munir, 2019):

$$P = m^{-1}(C - b)(mod\ y)$$

Contoh :

Misalkan *ciphertext* JMCJOJQ yang akan dibaca dan perlu didekripsi *ciphertext* tersebut dengan mengambil $m^{-1} = 21$ dan $b = 9$. Indeks karakter yang digunakan yaitu dimulai dari $\mathrm{A} = 0$ hingga $\mathrm{Z} = 26$. Perhitungan dekripsi adalah sebagai berikut:

Untuk huruf J = 9 maka kita dekripsikan

$$21(9 - 9) = 0 \equiv 0 \ (mod\ 26)$$

menjadi J→A

Tabel 2.3 Contoh Dekripsi *Affine Cipher*

Plaintext	J	M	C	J	O	J	Q
Indeks Karakter	9	12	2	9	14	9	16
$(21(C - 5))(mod\ 26)$	0	11	9	0	1	0	17
Ciphertext	A	L	J	A	B	A	R

2.1.10 Pembangkit Bilangan Acak

Bilangan acak (*random numbers*) merupakan elemen penting di dalam kriptografi yang dapat diartikan sebagai bilangan yang tidak dapat diprediksi kemunculannya maupun nilainya. Bilangan acak ada dua macam, yang pertama bilangan acak sejati (*true random numbers*) merupakan bilangan acak yang tidak dapat diulang pembangkitnya. Bilangan acak sejati dibangkitkan dari fenomena fisik yang terjadi secara acak. Misalkan seseorang menomori 100 potongan kertas dengan angka 1 sampai 100, lalu orang tersebut lemparkan semua kertas itu ke atas dan jatuh lagi ke lantai, kemudian perhatikan angka-angka yang muncul pada kertas yang tidak terbalik dan itu adalah angka-angka acak sejati. Jika dilemparkan kembali maka angka-angka yang mucul tidak sama dengan

percobaan pertama. Bilangan acak sejati bersifat *undeterministic*, artinya tidak dapat diprediksi kemunculannya dan tidak bisa diulang.

Bilangan acak yang kedua adalah bilangan semi acak (*pseudorandom numbers*). Bilangan acak jenis ini dibangkitkan dari sebuah prosedur (algoritma) komputasi berdasarkan satu atau lebih parameter nilai awal yang disebut umpan (*seed*) dan berlaku sebagai kunci (*key*). Pembangkit bilangan semi acak dinamakan *pseudorandom number generator* (*PRNG*). Sebuah *PNRG* dapat menghasilkan barisan bilangan acak yang panjang. Barisan bilangan acak yang sama dapat dibangkitkan ulang kembali asalkan kunci yang digunakan sama. Salah satu karakteristik pembangkit bilangan semi acak adalah sifat periodik, artinya barisan bilangan acak yang dibangkitkan akan berulang kembali dari awal setelah satu siklus atau satu periode. Panjang periode bergantung pada parameter awal yang digunakan. Karena sifat periodik itu maka bilangan semi acak bersifat deterministik (Munir, 2019).

2.1.11 Blum-Blum Shub Generator

Secara teoritis Blum-Blum Shub (BBS) termasuk *cryptographically secure pseudorandom generator* (*CSPRNG*) yang paling mudah dan paling efektif. Blum-Blum Shub dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum, dan Michael Shub. Untuk membangkitkan bilangan acak dengan Blum-Blum Shub, algoritmanya adalah sebagai berikut (Munir, 2019):

1. Pilih dua buah bilangan prima rahasia p dan q, yang masing-masing kongruen dengan 3 $(mod\ 4)$ (semakin besar bilangan prima tersebut maka semakin sulit dipecahkan oleh kriptanalis).

2. Kalikan keduanya menjadi $n = pq$. Bilangan n disebut Bilangan Bulat Blum.
3. Pilih bilangan bulat acak lain s sebagai umpan dengan syarat :

   a. $2 \leq s < n$

   b. $(s, n) = 1$

   Kemudian hitung $X_0 = s^2 \ (mod\ n)$
4. Barisan bit acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan :

   a. Hitung $x_i = {x_{i-1}}^2 \ (mod\ n)$

   b. $z_i = LSB$ (*Least Significant Bit*) dari x_i

   Barisan bit acak yang dihasilkan adalah z_1, z_2, z_3 ...

Bilangan acak tidak harus satu *least significant bit* tetapi bisa juga j buah bit (j adalah bilangan bulat positif yang tidak melebihi $(log_2(log_2\, n))$) (Munir, 2019). Nilai n disebut kunci publik dimana kunci tersebut dapat diumumkan kepada publik. Blum-Blum Shub tidak dapat diprediksi dari arah manapun, artinya jika diberikan barisan bit kriptanalis tidak dapat memprediksi barisan bit sebelumnya dan barisan bit sesudahnya (Schneier, B, 1996).

2.2 Kajian Integrasi Topik Dengan Al-Quran

Secara bahasa, amanah adalah jujur atau dapat dipercaya. Kata amanah berasal dari bahasa Arab yaitu *amina-amanatan* yang berarti pesan atau perintah. Pesan atau perintah yang dimaksud adalah sesuatu yang harus disampaikan dan dikerjakan orang lain dengan jujur. Secara istilah, amanah berarti segala suatu yang dipertanggung jawabkan kepada Allah baik berupa benda, perkataan,

perbuatan maupun kepercayaan (Andika, Muhammad, & Admizal, 2020). Sebagaimana dalam surah Al-Ahzab ayat 72 yang artinya :

"*Sesungguhnya Kami telah mengemukakan amanah kepada langit, bumi dan gunung-gunung, maka semuanya enggan untuk memikul amanah itu dan mereka khawatir akan mengkhianatinya, dan dipikullah amanah itu oleh manusia. Sesungguhnya manusia itu amat zalim dan amat bodoh*".

Dalam tafsir Ibnu Katsir (2017) 'Ali bin Abi Thalhah berkata dari Ibnu 'Abbas r.a : "Amanah adalah kewajiban-kewajiban yang ditawarkan oleh Allah kepada langit, bumi dan gunung-gunung. Jika mereka menunaikannya, Allah akan membalas mereka. Dan jika mereka menyia-menyiakannya, niscaya Allah akan menyiksa mereka. Mereka enggan menerimanya dan menolaknya bukan karena maksiat, akan tetapi karena *ta'zhim* (menghormati) agama Allah kalau-kalau mereka tidak mampu menunaikannya." (Katsir, 2017).

Konsep amanah telah dijelaskan pada ayat diatas bahwasanya setiap orang wajib menyampaikan amanah sesuai kodratnya. Hal ini sesuai dengan keamanan suatu pesan yang harus terjaga kerahasiannya. Isi pesan yang dikirim oleh pengirim pesan harus sesuai dengan isi pesan yang diterima oleh penerima pesan.

2.3 Kajian Topik dengan Teori Pendukung

Penelitian ini disusun menggunakan beberapa teori pendukung, seperti *Affine Cipher* yang temasuk kriptografi simetris. *Affine Cipher* ini merupakan Cipher yang dapat mengenkripsi pesan dengan cara mengalikan indeks karakter *plaintext P* dengan kunci multiplikatif m kemudian menambahkan hasil tersebut dengan nilai pergeseran aditif b. Pada kunci multiplikatif membahas mengenai teori bilangan meliputi keterbagian, dan pembagi bersama terbesar. *Affine Cipher* juga dapat mendekripsi pesan dengan cara mengurangi indeks karakter *ciphertext*

dengan nilai pergeseran aditif b kemudian dikalikan dengan balikan modulo dari kunci multiplikatif m. Untuk balikan modulo dari kunci multiplikatif m ini membahas tentang keterbagian, pembagi bersama terbesar, balikan modulo, algoritma euclid, dan kombinasi linier. Sedangkan nilai pergeseran aditif pada *Affine Cipher* yang telah dimodifikasi oleh algoritma Blum-Blum Shub membahas beberapa teori meliputi kekongruenan, bilangan biner, dan *least significant bit* (*lsb*).

BAB III
METODE PENELITIAN

3.1 Jenis Penelitian

Jenis penelitian yang digunakan adalah studi literatur yaitu kegiatan dalam penelitian yang bertujuan untuk mengembangkan aspek teoritis maupun aspek manfaat praktis dengan menemukan referensi dan hasil-hasil riset yang berkaitan dengan bidang ilmu. Data yang dikumpulkan dalam penelitian ini bersifat deskriptif dalam bentuk kata-kata seperti jurnal, laporan hasil penelitian, buku yang relevan, artikel, skripsi dan lain sebagainya yang mendukung penelitian ini.

3.2 Pra Penelitian

Proses pra penelitian ini diawali dengan mencari referensi penelitian dimana peneliti mencari sumber penelitian dari jurnal, artikel, buku yang relevan serta semacamnya. Kemudian, peneliti menganalisis suatu masalah untuk mendapatkan algoritma yang sesuai. Selanjutnya, peneliti merancang algoritma yang akan digunakan pada proses penelitian selanjutnya.

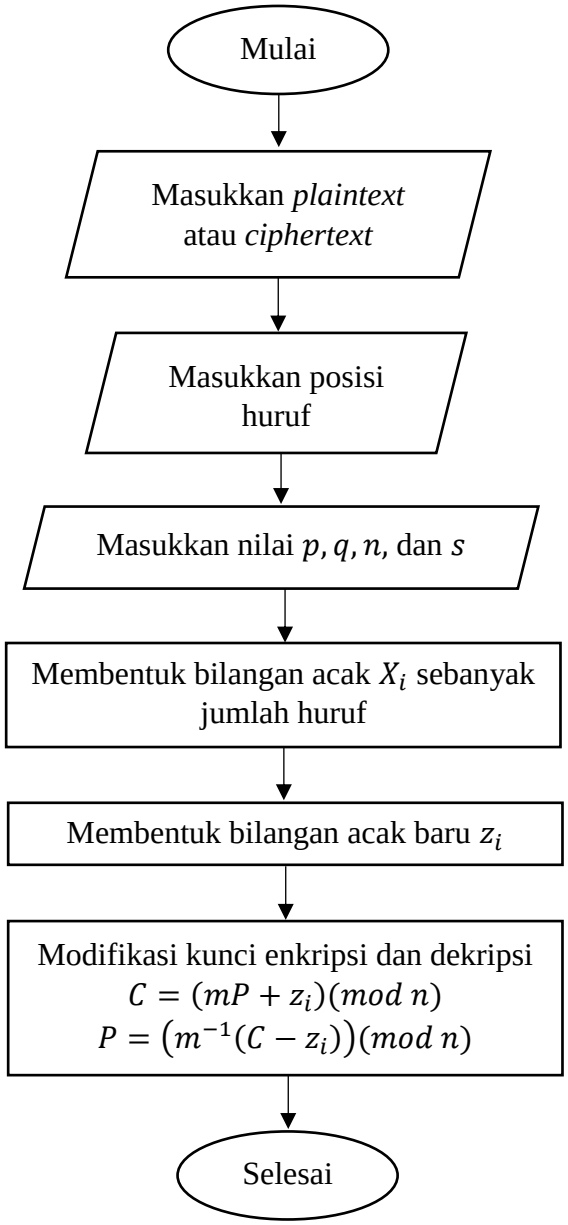
3.3 Tahapan Penelitian

Penelitian ini memiliki 3 tahapan diantaranya tahap modifikasi, tahap enkripsi, dan tahap dekripsi, dimana pada tahap modifikasi Blum-Blum Shub dibangkitkan agar menghasilkan bilangan acak yang nantinya digunakan sebagai pengganti nilai pergeseran aditif pada *Affine Cipher* dan pada tahap enkripsi mengubah *plaintext* menjadi *ciphertext*. Sedangkan, tahap dekripsi mengubah *ciphertext* menjadi *plaintext*. Berikut langkah-langkah yang digunakan pada tahap penelitian modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub :

1. Proses modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub:
   a. Menentukan *plaintext* atau *ciphertext* serta menentukan posisi huruf tersebut.
   b. Menentukan nilai p dan q, kemudian kalikan keduanya sehingga menghasilkan nilai n.
   c. Menentukan nilai s (*seed*) kemudian hitung $X_0 = s^2 \ (mod\ n)$.
   d. Membentuk bilangan acak $X_i = {X_{i-1}}^2 \ (mod\ n)$ dan lakukan iterasi sebanyak jumlah posisi huruf *plaintext*.
   e. Membentuk bilangan acak baru z_i.
   f. Melakukan modifikasi kunci enkripsi dan kunci dekripsi *Affine Cipher* dengan mengubah nilai b menjadi z_i.

      Modifikasi kunci enkripsi *Affine Cipher* : $C = (mP + z_i)(mod\ n)$

      Modifikasi kunci dekripsi *Affine Cipher* : $P = (m^{-1}(C - z_i))(mod\ n)$



Gambar 3.1 *Flowchart* Modifikasi *Affine Cipher*

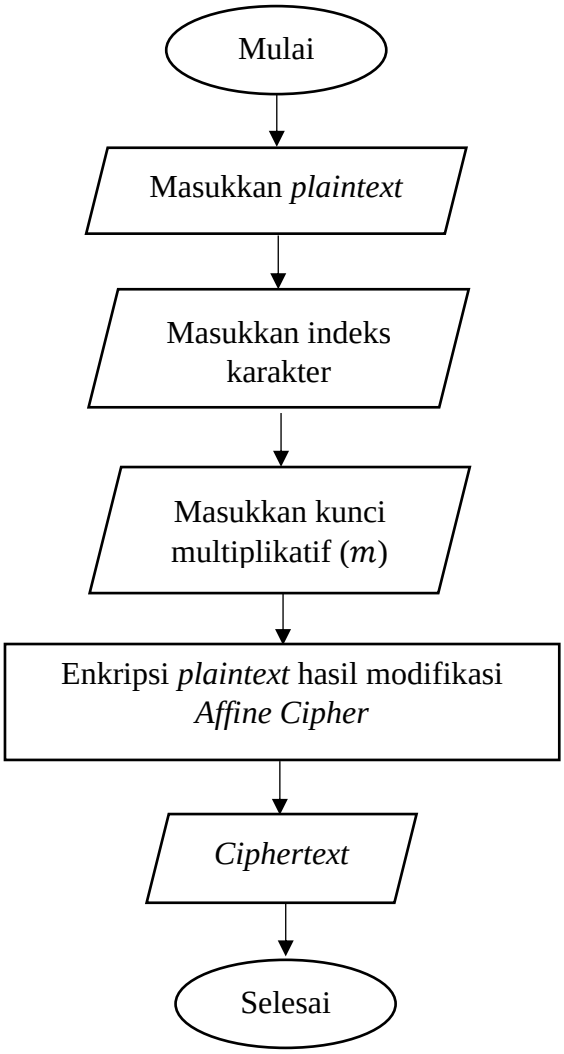
2. Proses enkripsi hasil modifikasi *affine cipher* dan algoritma Blum-Blum Shub:

   a. Menulis ulang pesan teks asli (*plaintext*).

   b. Menetukan indeks karakter dari *plaintext* tersebut.

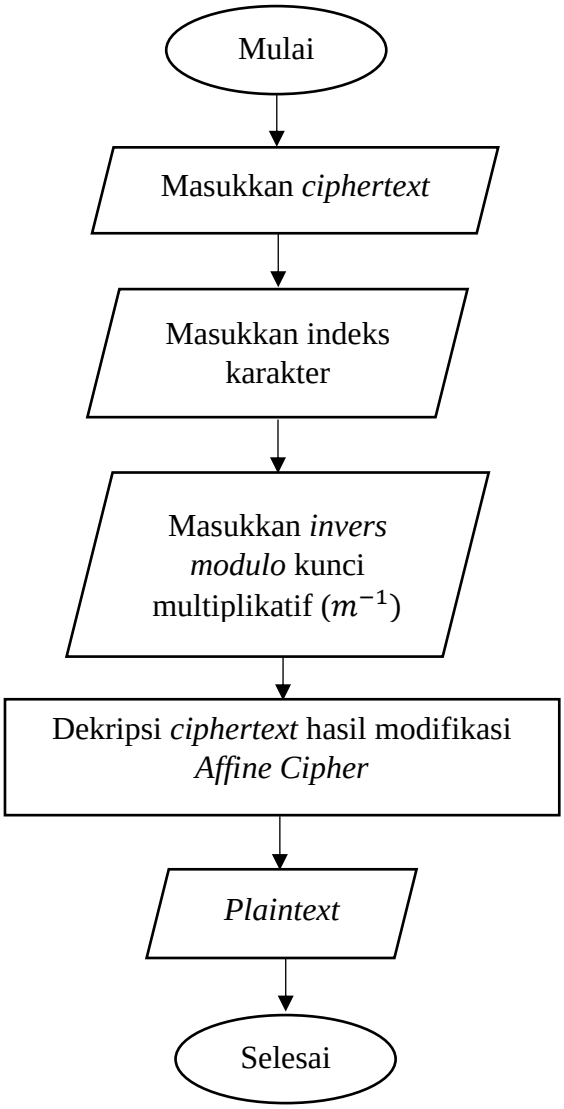
   c. Menentukan kunci multiplikatif (m).

   d. Melakukan proses enkripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub.



Gambar 3.2 *Flowchart* Enkripsi Hasil Modifikasi *Affine Cipher*

3. Proses dekripsi hasil modifikasi *Affine Cipher* dan algoritma Blum-Blum Shub:
   a. Menulis ulang pesan rahasia (*ciphertext*).
   b. Menetukan indeks karakter dari *ciphertext* tersebut.
   c. Menentukan balikan modulo dari kunci multiplikatif (m).
   4. Melakukan proses dekripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub.



Gambar 3.3 *Flowchart* Dekripsi Hasil Modifikasi *Affine Cipher*

BAB IV
HASIL DAN PEMBAHASAN

3.1 Proses Modifikasi *Affine Cipher* menggunakan Algoritma Blum - Blum Shub

Proses modifikasi *Affine Cipher* ini dimulai dengan membangkitkan bilangan acak yang dihasilkan dari algoritma Blum-Blum Shub. Beberapa langkah yang harus dilakukan dalam proses modifikasi ini yaitu :

1. Menentukan pesan teks asli (*plaintext*) serta menentukan posisi huruf *plaintext* tersebut.

Tabel 4.1 Posisi *Plaintext*

Plaintext	S	y	a	h	l	a	Space
Posisi	1	2	3	4	5	6	7
Plaintext	A	z	n	a	d	i	l
Posisi	8	9	10	11	12	13	14
Plaintext	l	a	h	-	1	8	6
Posisi	15	16	17	18	19	20	21
Plaintext	1	0	0	5	4		
Posisi	22	23	24	25	26		

2. Menentukan nilai p dan q, kemudian kalikan keduanya sehingga menghasilkan nilai n. Dipilih nilai $p = 6427$ dan $q = 8999$ sehingga diperoleh nilai modulus $n = pq = 57836573$.
3. Menentukan nilai s (*seed*) kemudian hitung $X_0 = s^2 (mod\ n)$. Nilai umpan (*seed*) yang dipilih yaitu $s = 2221$ maka $X_0 = 2221^2\ (mod\ 57836573) = 4932841$.
4. Membentuk bilangan acak $X_i = {X_{i-1}}^2\ (mod\ n)$ dan lakukan iterasi sebanyak jumlah posisi huruf *plaintext*.

Tabel 4.2 Bilangan Acak X_i

Barisan bilangan acak	**Proses Algoritma *Blum-Blum Shub* $(X_{i-1}{}^2 \ mod \ n)$**	**Hasil**
X_1	$4932841^2 \ (mod \ 57836573)$	33011867
X_2	$33011867^2 \ (mod \ 57836573)$	49536109
X_3	$49536109^2 \ (mod \ 57836573)$	702192
X_4	$702192^2 \ (mod \ 57836573)$	16820039
X_5	$16820039^2 \ (mod \ 57836573)$	42291856
X_6	$42291856^2 \ (mod \ 57836573)$	32117885
X_7	$32117885^2 \ (mod \ 57836573)$	53498194
X_8	$53498194^2 \ (mod \ 57836573)$	7742543
X_9	$7742543^2 \ (mod \ 57836573)$	394652
X_{10}	$394652^2 \ (mod \ 57836573)$	54146588
X_{11}	$54146588^2 \ (mod \ 57836573)$	45447992
X_{12}	$45447992^2 \ (mod \ 57836573)$	16146998
X_{13}	$16146998^2 \ (mod \ 57836573)$	8425194
X_{14}	$8425194^2 \ (mod \ 57836573)$	26836422
X_{15}	$26836422^2 \ (mod \ 57836573)$	46066316
X_{16}	$46066316^2 \ (mod \ 57836573)$	56873926
X_{17}	$56873926^2 \ (mod \ 57836573)$	31674003
X_{18}	$31674003^2 \ (mod \ 57836573)$	16934329
X_{19}	$16934329^2 \ (mod \ 57836573)$	13918330
X_{20}	$13918330^2 \ (mod \ 57836573)$	10266072
X_{21}	$10266072^2 \ (mod \ 57836573)$	1852518
X_{22}	$1852518^2 \ (mod \ 57836573)$	32044796
X_{23}	$32044796^2 \ (mod \ 57836573)$	30155144
X_{24}	$30155144^2 \ (mod \ 57836573)$	24660313
X_{25}	$24660313^2 \ (mod \ 57836573)$	4146384
X_{26}	$4146384^2 \ (mod \ 57836573)$	54567

5. Membentuk bilangan acak baru z_i, karena bilangan acak baru z_i tidak harus satu *least significant bit* maka peneliti memilih j = 4 (j tidak melebihi $log_2(log_2\ 57836573) = 4{,}68849$).

   1. $X_1 = 33011867 = 1111101111011100010011011$

      $z_1 = 33011867 = 1011_{basis\ 2} \equiv 11\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 33011867.

   2. $X_2 = 49536109 = 10111100111101110001101101$

      $z_2 = 49536109 = 1101_{basis\ 2} \equiv 13\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 49536109.

   3. $X_3 = 702192 = 10101011011011110000$

      $z_3 = 702192 = 0000_{basis\ 2} \equiv 0\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 702192.

   4. $X_4 = 16820039 = 1000000001010011101000111$

      $z_4 = 16820039 = 0111_{basis\ 2} \equiv 7\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 16820039.

   5. $X_5 = 42291856 = 10100001010101001010010000$

      $z_5 = 42291856 = 0000_{basis\ 2} \equiv 0\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 42291856.

   6. $X_6 = 32117885 = 11110101000010100011111101$

      $z_6 = 32117885 = 1101_{basis\ 2} \equiv 13\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 32117885.

   7. $X_7 = 53498194 = 11001100000101000101010010$

      $z_7 = 53498194 = 0010_{basis\ 2} \equiv 2\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 53498194.

8. $X_8 = 7742543 = 111011000100100010011111$

   $z_8 = 7742543 = 1111_{basis\ 2} \equiv 15\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 7742543.

9. $X_9 = 394652 = 1100000010110011100$

   $z_9 = 394652 = 1100_{basis\ 2} \equiv 12\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 394652.

10. $X_{10} = 54146588 = 11001110100011011000011100$

    $z_{10} = 54146588 = 1100_{basis\ 2} \equiv 12\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 54146588.

11. $X_{11} = 45447992 = 10101101010111101100111000$

    $z_{11} = 45447992 = 1000_{basis\ 2} \equiv 8\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 45447992.

12. $X_{12} = 16146998 = 111101100110001000110110$

    $z_{12} = 16146998 = 0110_{basis\ 2} \equiv 6\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 16146998.

13. $X_{13} = 8425194 = 100000001000111011101010$

    $z_{13} = 8425194 = 1010_{basis\ 2} \equiv 10\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 8425194.

14. $X_{14} = 26836422 = 110011001011111010111000110$

    $z_{14} = 26836422 = 0110_{basis\ 2} \equiv 6\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 26836422.

15. $X_{15} = 46066316 = 10101111101110101010001100$

    $z_{15} = 46066316 = 1100_{basis\ 2} \equiv 12\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 46066316.

16. $X_{16} = 56873926 = 11011000111101001111000110$

    $z_{16} = 56873926 = 0110_{basis\ 2} \equiv 6\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 56873926.

17. $X_{17} = 31674003 = 11110001101001110100100 11$

    $z_{17} = 31674003 = 0011_{basis\ 2} \equiv 3\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 31674003.

18. $X_{18} = 16934329 = 10000001001100101101110 01$

    $z_{18} = 16934329 = 1001_{basis\ 2} \equiv 9\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 16934329.

19. $X_{19} = 13918330 = 11010100011000000111101 0$

    $z_{19} = 13918330 = 1010_{basis\ 2} \equiv 10\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 13918330.

20. $X_{20} = 10266072 = 10011100101001011101 1000$

    $z_{20} = 10266072 = 1000_{basis\ 2} \equiv 8\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 10266072.

21. $X_{21} = 1852518 = 111000100010001100110$

    $z_{21} = 1852518 = 0110_{basis\ 2} \equiv 6\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 1852518.

22. $X_{22} = 32044796 = 11110100011110111011111 100$

    $z_{22} = 32044796 = 1100_{basis\ 2} \equiv 12\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 32044796.

23. $X_{23} = 30155144 = 11100110000100001100010 00$

    $z_{23} = 30155144 = 1000_{basis\ 2} \equiv 8\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 30155144.

24. $X_{24} = 24660313 = 10111100001001001010110 01$

$z_{24} = 24660313 = 1001_{basis\ 2} \equiv 9\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 24660313.

25. $X_{25} = 4146384 = 11111101000100110100 00$

$z_{25} = 4146384 = 0000_{basis\ 2} \equiv 0\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 4146384.

26. $X_{26} = 585476 = 10001110111100000100$

$z_{26} = 585476 = 0100_{basis\ 2} \equiv 4\ (mod\ 2^4)$ diambil 4 bit *LSB* dari 585476.

Jadi, barisan blok bit acak baru (z_i) yang dihasilkan adalah 11, 13, 0, 7, 0, 13, 2, 15, 12, 12, 8, 6, 10, 6, 12, 6, 10, 6, 12, 6, 3, 9, 10, 8, 6, 12, 8, 9, 0, 4.

6. Melakukan modifikasi kunci enkripsi dan kunci dekripsi *Affine Cipher* dengan mengubah nilai b menjadi z_i.

Modifikasi kunci enkripsi *Affine Cipher* : $C = (mP + z_i)(mod\ y)$.

Tabel 4.3 Kunci Enkripsi Hasil Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub

Posisi huruf ke-i	Bilangan z_i	Kunci Enkripsi Hasil Modifikasi
1	11	$(m(P) + 11) \ (mod\ 256)$
2	13	$(m(P) + 13) \ (mod\ 256)$
3	0	$(m(P) + 0) \ (mod\ 256)$
4	7	$(m(P) + 7) \ (mod\ 256)$
5	0	$(m(P) + 0) \ (mod\ 256)$
6	13	$(m(P) + 13) \ (mod\ 256)$
7	2	$(m(P) + 2) \ (mod\ 256)$
8	15	$(m(P) + 15) \ (mod\ 256)$
9	12	$(m(P) + 12) \ (mod\ 256)$
10	12	$(m(P) + 12) \ (mod\ 256)$
11	8	$(m(P) + 8) \ (mod\ 256)$
12	6	$(m(P) + 6) \ (mod\ 256)$
13	10	$(m(P) + 10) \ (mod\ 256)$
14	6	$(m(P) + 6) \ (mod\ 256)$
15	12	$(m(P) + 12) \ (mod\ 256)$
16	6	$(m(P) + 6) \ (mod\ 256)$
17	3	$(m(P) + 3) \ (mod\ 256)$
18	9	$(m(P) + 9) \ (mod\ 256)$
19	10	$(m(P) + 10) \ (mod\ 256)$
20	8	$(m(P) + 8) \ (mod\ 256)$
21	6	$(m(P) + 6) \ (mod\ 256)$
22	12	$(m(P) + 12) \ (mod\ 256)$
23	8	$(m(P) + 8) \ (mod\ 256)$
24	9	$(m(P) + 9) \ (mod\ 256)$
25	0	$(m(P) + 0) \ (mod\ 256)$
26	4	$(m(P) + 4) \ (mod\ 256)$

Modifikasi kunci dekripsi *Affine Cipher* : $P = (m^{-1}(C - z_i))(mod\ y)$.

Tabel 4.4 Kunci Dekripsi Hasil Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub

Posisi huruf ke-i	Bilangan z_i	Kunci Dekripsi Hasil Modifikasi
1	11	$(m^{-1}(C - 11))(mod\ 256)$
2	13	$(m^{-1}(C - 13))(mod\ 256)$
3	0	$(m^{-1}(C - 0))(mod\ 256)$
4	7	$(m^{-1}(C - 7))(mod\ 256)$
5	0	$(m^{-1}(C - 0))(mod\ 256)$
6	13	$(m^{-1}(C - 13))(mod\ 256)$
7	2	$(m^{-1}(C - 2))(mod\ 256)$
8	15	$(m^{-1}(C - 15))(mod\ 256)$
9	12	$(m^{-1}(C - 12))(mod\ 256)$
10	12	$(m^{-1}(C - 12))(mod\ 256)$
11	8	$(m^{-1}(C - 8))(mod\ 256)$
12	6	$(m^{-1}(C - 6))(mod\ 256)$
13	10	$(m^{-1}(C - 10))(mod\ 256)$
14	6	$(m^{-1}(C - 6))(mod\ 256)$
15	12	$(m^{-1}(C - 12))(mod\ 256)$
16	6	$(m^{-1}(C - 6))(mod\ 256)$
17	3	$(m^{-1}(C - 3))(mod\ 256)$
18	9	$(m^{-1}(C - 9))(mod\ 256)$
19	10	$(m^{-1}(C - 10))(mod\ 256)$
20	8	$(m^{-1}(C - 8))(mod\ 256)$
21	6	$(m^{-1}(C - 6))(mod\ 256)$
22	12	$(m^{-1}(C - 12))(mod\ 256)$
23	8	$(m^{-1}(C - 8))(mod\ 256)$
24	9	$(m^{-1}(C - 9))(mod\ 256)$
25	0	$(m^{-1}(C - 0))(mod\ 256)$
26	4	$(m^{-1}(C - 4))(mod\ 256)$

4.2 Proses Enkripsi Hasil Modifikasi *Affine Cipher* Menggunakan Algoritma Blum - Blum Shub

Langkah-langkah enkripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum - Blum Shub adalah sebagai berikut :

1. Menentukan pesan teks asli (*plaintext*) serta menentukan indeks karakter. Peneliti melakukan proses enkripsi menggunakan pesan asli (*plaintext*)-nya adalah **Syahla Aznadillah-18610054**. Peneliti memilih *Plaintext* tersebut karena berdasarkan nama lengkap serta nomor induk mahasiswa (NIM) dari peneliti. *Plaintext* tersebut bisa dirubah sesuai dengan kebutuhan dan keinginan penelitian selanjutnya.
2. Menentukan indeks karakter dari *plainteks*.

Tabel 4.5 Indeks Karakter *Plaintext*

Plaintext	S	y	a	h	l	a	Space
Indeks Karakter	83	121	97	104	108	97	32
Plaintext	A	z	n	a	d	i	l
Indeks Karakter	65	122	110	97	100	105	108
Plaintext	l	a	h	-	1	8	6
Indeks Karakter	108	97	104	45	49	56	54
Plaintext	1	0	0	5	4		
Indeks Karakter	49	48	48	53	52		

3. Menentukan kunci multiplikatif (m).

   Peneliti memilih nilai $m = 29$ karena $(256, 29) = 1$.

4. Melakukan proses enkripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub.

Tabel 4.6 Enkripsi *Plaintext* Hasil Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub

Indeks *Plaintext*	**Bilangan Acak Baru (z_i)**	**Enkripsi *Affine Cipher* $((29P + z_i)(mod\ 256))$**	**Indeks *Ciphertext***	***Ciphertext***
83	11	$(29(83) + 11)(mod\ 256)$	114	r
121	13	$(29(121) + 13)(mod\ 256)$	194	┬
97	0	$(29(97) + 0)\ (mod\ 256)$	253	²
104	7	$(29(104) + 7)(mod\ 256)$	207	¤
108	0	$(29(108) + 0)(mod\ 256)$	60	<
97	13	$(29(97) + 13)(mod\ 256)$	10	LF
32	2	$(29(32) + 2)(mod\ 256)$	162	ó
65	15	$(29(65) + 15)(mod\ 256)$	108	l
122	12	$(29(122) + 12)(mod\ 256)$	222	ì
110	12	$(29(110) + 12)(mod\ 256)$	130	é
97	8	$(29(97) + 8)(mod\ 256)$	5	ENQ
100	6	$(29(100) + 6)(mod\ 256)$	90	Z
105	10	$(29(105) + 10)(mod\ 256)$	239	´
108	6	$(29(108) + 6)(mod\ 256)$	66	B
108	12	$(29(108) + 12)(mod\ 256)$	72	H
97	6	$(29(97) + 6)(mod\ 256)$	3	ETX
104	3	$(29(104) + 3)(mod\ 256)$	203	╦
45	9	$(29(45) + 9)(mod\ 256)$	34	"
49	10	$(29(49) + 10)(mod\ 256)$	151	ù
56	8	$(29(56) + 8)(mod\ 256)$	96	`
54	6	$(29(54) + 6)(mod\ 256)$	36	$
49	12	$(29(49) + 12)(mod\ 256)$	153	Ö
48	8	$(29(48) + 8)(mod\ 256)$	120	x
48	9	$(29(48) + 9)(mod\ 256)$	121	y
53	0	$(29(53) + 0)(mod\ 256)$	1	SOH
52	4	$(29(52) + 4)(mod\ 256)$	232	Þ

Ciphertext yang dihasilkan adalah r┬²¤<LFólìéENQZ´BHETX╦"ù`$ÖxySOHÞ.

4.3 Proses Dekripsi Hasil Modifikasi *Affine Cipher* Menggunakan Algoritma Blum - Blum Shub

Langkah-langkah dekripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum - Blum Shub adalah sebagai berikut :

1. a Penelitian kali ini, peneliti melakukan proses dekripsi menggunakan hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub. Pesan rahasia (*ciphertext*) yang akan diubah menjadi pesan asli (*plaintext*). *Ciphertext* yang akan digunakan adalah r┬²¤<LFólìéENQZ´BHETX╦"ù`$ÖxySOHÞ.k
2. Menentukan indeks karakter dari *ciphertext*.

Tabel 4.7 Indeks Karakter *Ciphertext*

Ciphertext	r	┬	²	¤	<	LF	ó
Indeks Karakter	114	194	253	207	60	10	162
Ciphertext	l	ì	é	ENQ	Z	´	B
Indeks Karakter	108	222	130	5	90	239	66
Ciphertext	H	ETX	╦	"	ù	`	$
Indeks Karakter	72	3	203	34	151	96	36
Ciphertext	Ö	x	y	SOH	Þ		
Indeks Karakter	153	120	121	1	232		

3. Menentukan balikan modulo dari kunci multiplikatif (m^{-1}). Untuk menentukan m^{-1} peneliti terlebih dahulu menentukan balikan modulo dari 29 ($mod\ 256$) menggunakan algoritma Euclid dimana $a = 256, b = 29$. Maka balikan modulo yang dihitung dengan algoritma Euclid sebagai berikut:

$$256 = 29 \times 8 + 24$$

$$29 = 24 \times 1 + 5$$

$$24 = 5 \times 4 + 4$$

$$5 = 4 \times 1 + 1$$

$$4 = 1 \times 4$$

Karena $(256,29) = 1$, tulis 1 sebagai sebuah kombinasi linier dari 256 dan 29 maka:

$$\begin{aligned} 1 &= 5 - 4 \times 1 \\ &= 5 - [24 - 5 \times 4] \times 1 \\ &= -24 + 5 \times 5 \\ &= -24 + [29 - 24 \times 1] \times 5 \\ &= 29 \times 5 - 24 \times 6 \\ &= 29 \times 5 - [256 - 29 \times 8] \times 6 \\ &= -256 \times 6 + 29 \times \mathbf{53} \end{aligned}$$

Didapat *invers modulo* dari 29 (mod 256) yaitu 53.

Jadi, balikan modulo dari kunci multiplikatif (m) yaitu $m^{-1} = 53$.

4. Melakukan proses dekripsi hasil modifikasi *Affine Cipher* menggunakan algoritma Blum-Blum Shub.

Tabel 4.8 Dekripsi *Ciphertext* Hasil Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub

Indeks *Ciphertext*	**Bilangan Acak Baru (z_i)**	**Rumus dekripsi *Affine Cipher*** $(m^{-1}(C - z_i)(mod\ 256))$	**Indeks *Plaintext***	***Plaintext***
114	11	$53(114 - 11)(mod\ 256)$	83	S
194	13	$53(194 - 13)(mod\ 256)$	121	y
253	0	$53(253 - 0)(mod\ 256)$	97	a
207	7	$53(207 - 7)(mod\ 256)$	104	h
60	0	$53(60 - 0)\ (mod\ 256)$	108	l
10	13	$53(10 - 13)(mod\ 256)$	97	a
162	2	$53(162 - 2)(mod\ 256)$	32	Space
108	15	$53(108 - 15)(mod\ 256)$	65	A
222	12	$53(222 - 12)(mod\ 256)$	122	z
130	12	$53(130 - 12)(mod\ 256)$	110	n
5	8	$53(5 - 8)(mod\ 256)$	97	a
90	6	$53(90 - 6)(mod\ 256)$	100	d
239	10	$53(239 - 10)(mod\ 256)$	105	i
66	6	$53(66 - 6)(mod\ 256)$	108	l
72	12	$53(72 - 12)(mod\ 256)$	108	l
3	6	$53(3 - 6)(mod\ 256)$	97	a
203	3	$53(203 - 3)(mod\ 256)$	104	h
34	9	$53(34 - 9)(mod\ 256)$	45	-
151	10	$53(151 - 10)(mod\ 256)$	49	1
96	8	$53(96 - 8)(mod\ 256)$	56	8
36	6	$53(36 - 6)(mod\ 256)$	54	6
153	12	$53(153 - 12)(mod\ 256)$	49	1
120	8	$53(120 - 8)(mod\ 256)$	48	0
121	9	$53(121 - 9)(mod\ 256)$	48	0
1	0	$53(1 - 0)(mod\ 256)$	53	5
232	4	$53(232 - 4)(mod\ 256)$	52	4

Plaintext yang dihasilkan adalah Syahla Aznadillah-18610054.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil dan pembahasan di atas, dapat diambil beberapa kesimpulan :

1. Proses modifikasi *Affine Cipher* menggunakan Algoritma Blum-Blum Shub diawali dengan memilih nilai $p, q,$ dan s. Selanjutnya membentuk bilangan acak X_i serta melakukan iterasi sebanyak jumlah huruf yang terdapat di *plaintext* atau *ciphertext* kemudian membentuk bilangan acak baru z_i. Bilangan acak baru z_i yang dihasilkan akan mengubah nilai pergeseran aditif (b) pada *Affine Cipher*. Kemudian kunci hasil modifikasi tersebut dapat mengenkripsi *plaintext* atau mendekripsi *ciphertext*.
2. Proses enkripsi hasil modifikasi *Affine Cipher* menggunakan Algoritma Blum-Blum Shub diawali dengan menentukan kunci multiplikatif (m). Selanjutnya, memasukkan indeks karakter setiap huruf *plaintext* kedalam kunci enkripsi hasil modifikasi *Affine Cipher* menggunakan Algoritma Blum-Blum Shub. Kemudian lakukan perhitungan sehingga menghasilkan *ciphertext*.
3. Proses dekripsi hasil modifikasi *Affine Cipher* menggunakan Algoritma Blum-Blum Shub diawali dengan menentukan balikan modulo kunci multiplikatif (m^{-1}). Selanjutnya, memasukkan indeks karakter setiap huruf *ciphertext* kedalam kunci dekripsi hasil modifikasi *Affine Cipher* menggunakan Algoritma Blum-Blum Shub. Kemudian lakukan perhitungan sehingga menghasilkan *plaintext*.

5.2 Saran untuk Penelitian Lanjutan

Penelitian ini membahas mengenai modifikasi *Affine Cipher* menggunakan Algoritma Blum-Blum Shub. Untuk penelitian selanjutnya disarankan menggunakan nilai yang lebih tinggi, penambahan karakter baru serta menggunakan seluruh karakter ASCII pada proses enkripsi dan dekripsi. Selain itu, penelitian selanjutnya disarankan membuat suatu modifikasi algoritma lainnya yang lebih mudah digunakan serta tingkat keamanannya lebih tinggi atau dapat juga menggunakan aplikasi program komputer yang lain.

DAFTAR PUSTAKA

Al Qur'an dan Terjemahan. (2019). Kementerian Agama RI.

Andika, T., Muhammad, T., & Admizal, I. (2020). Amanah dan Khianat dalam Al-Qur'an Menurut Quraish Shihab. *Al Tadabbur: Jurnal Ilmu Al-Qur'an dan Tafsir Vol: 05 No. 02 November 2020*, 182.

Ariyus, D. (2006). *Kriptografi Keamanan Data dan Komunikasi.* Yogyakarta: Graha Ilmu.

Batten, L. M. (2012). *Public Key Cryptography Applications and Attacks.* Netherlands: IEEE Press.

Borodzieva, & Manoilov. (2014). Training Module with Graphical User Interface for Encryption and Decryption Using Affine Ciphers Applied in Cryptosystems. *IEEE 20th International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 281-286.

Das, A. (2013). *Computational Number Theory.* Boca Raton: Taylor & Francis Group.

Irawan, W. H. (2014). *Pengantar Teori Bilangan.* Malang: UIN-Malang Press.

Jamaludin, & Romindo. (2020). *Kriptografi Teknik Hybrid Criptosystem Menggunakan Kombinasi Vigenere Cipher dan RSA.* Yayasan Kita Menulis.

Katsir, I. (2017). *Tafsir Ibnu Katsir Jilid 7.* Pustaka Imam Asy-Syafi'i.

Kraft, J., & Washington, L. (2015). *Elementary Number Theory.* New York: Taylor & Francis Group.

Kromodimoeljo, S. (2010). *Teori dan Aplikasi Kriptografi.* Jakarta: SPK IT Consuling.

Laia, O. (2019). Application of Linear Congruent Generator in Affine Cipher Algorithm to Produce Dynamic Encryption. *Journal of Physics: Conference Series (Vol. 1361, No. 1, p. 012001).*

Munir, R. (2019). *Kriptografi edisi dua.* Bandung: Informatika Bandung.

Sadikin, R. (2012). *Kriptografi untuk Keamanan Jaringan.* Yogyakarta: CV Andi Offset.

Schneier, B. (1996). *Applied Cryptography 2nd.* John Wiley Sons.

Stallings, W. (2003). *Cryptography and Network Security.* New Jersey: Pearson Education.

LAMPIRAN

Lampiran 1. Program Pyhton Algoritma Blum-Blum Shub

```
def blum_blum_shub(x, M):
    return (x*x)%M
P=6427
Q=8999
M=P*Q

seed=2221
x=(seed**2)%M

for _ in range(26):
    x=blum_blum_shub(x, M)
    biner=bin(blum_blum_shub(x, M))
    w=x%2**4
    print(w)
```

Lampiran 2. Tabel ASCII

Indeks Karakter	Symbol	Description
000	NULL	(Null Character)
001	SOH	(Start Of Header)
002	STX	(Start Of Text)
003	ETX	(End Of Text)
004	EOT	(End Of Trans)
005	ENQ	(Enquiry)
006	ACK	(Acknowledgement)
007	BEL	(Bell)
008	BS	(Backspace)
009	HT	(Horizontal Tab)
010	LF	(Line Feed)
011	VT	(Vertical Tab)
012	FF	(Form Feed)
013	CR	(Carriage Return)
014	SO	(Shift Out)
015	SI	(Shift In)
016	DLE	(Data Link Escape)
017	DC1	(Device Control 1)
018	DC2	(Device Control 2)
019	DC3	(Device Control 3)
020	DC4	(Device Control 4)
021	NAK	(Negative Acknowl)
022	SYN	(Synchronous Idle)
023	ETB	(End Of Trans. Block)
024	CAN	(Cancel)
025	EM	(End Of Medium)
026	SUB	(Substitute)
027	ESC	(Escape)

Indeks Karakter	Symbol	Description
028	FS	(File Separator)
029	GS	(Group Separator)
030	RS	(Record Separator)
031	US	(Unit Separator)
032	Space	(Space)
033	!	(Exclamation Mark)
034	"	(Quotation Mark)
035	#	(Number Sign)
036	$	(Dollar Sign)
037	%	(Percent Sign)
038	&	(Ampersand)
039	'	(Apostrophe)
040	(	(Round Brackets)
041	)	(Round Brackets)
042	*	(Asterisk)
043	+	(Plus Sign)
044	,	(Comma)
045	-	(Hyphen)
046	.	(Dot, Full Stop)
047	/	(Slash)
048	0	(Number Zero)
049	1	(Number One)
050	2	(Number Two)
051	3	(Number Three)
052	4	(Number Four)
053	5	(Number Five)
054	6	(Number Six)
055	7	(Number Seven)
056	8	(Number Eight)
057	9	(Number Nine)

Indeks Karakter	Symbol	Description
058	:	(Colon)
059	;	(Semicolon)
060	<	(Less-Than Sign)
061	=	(Equals Sign)
062	>	(Greater-Than Sign)
063	?	(Question Mark)
064	@	(At Sign)
065	A	(Capital Letter A)
066	B	(Capital Letter B)
067	C	(Capital Letter C)
068	D	(Capital Letter D)
069	E	(Capital Letter E)
070	F	(Capital Letter F)
071	G	(Capital Letter G)
072	H	(Capital Letter H)
073	I	(Capital Letter I)
074	J	(Capital Letter J)
075	K	(Capital Letter K)
076	L	(Capital Letter L)
077	M	(Capital Letter M)
078	N	(Capital Letter N)
079	O	(Capital Letter O)
080	P	(Capital Letter P)
081	Q	(Capital Letter Q)
082	R	(Capital Letter R)
083	S	(Capital Letter S)
084	T	(Capital Letter T)
085	U	(Capital Letter U)
086	V	(Capital Letter V)
087	W	(Capital Letter W)

Indeks Karakter	Symbol	Description
088	X	(Capital Letter X)
089	Y	(Capital Letter Y)
090	Z	(Capital Letter Z)
091	[	(Square Brackets)
092	\	(Backslash)
093	]	(Square Brackets)
094	^	(Circumflex Accent Or Caret)
095	_	(Underscore Or Understrike)
096	`	(Grave Accent)
097	a	(Lowercase Letter A)
098	b	(Lowercase Letter B)
099	c	(Lowercase Letter C)
100	d	(Lowercase Letter D)
101	e	(Lowercase Letter E)
102	f	(Lowercase Letter F)
103	g	(Lowercase Letter G)
104	h	(Lowercase Letter H)
105	i	(Lowercase Letter I)
106	j	(Lowercase Letter J)
107	k	(Lowercase Letter K)
108	l	(Lowercase Letter L)
109	m	(Lowercase Letter M)
110	n	(Lowercase Letter N)
111	o	(Lowercase Letter O)
112	p	(Lowercase Letter P)
113	q	(Lowercase Letter Q)
114	r	(Lowercase Letter R)
115	s	(Lowercase Letter S)
116	t	(Lowercase Letter T)
117	u	(Lowercase Letter U)

Indeks Karakter	Symbol	Description
118	v	(Lowercase Letter V)
119	w	(Lowercase Letter W)
120	x	(Lowercase Letter X)
121	y	(Lowercase Letter Y)
122	z	(Lowercase Letter Z)
123	{	(Curly Brackets Or Braces)
124	|	(Vertical Bar Or Vertical Slash)
125	}	(Curly Brackets Or Braces)
126	~	(Tilde Or Swung Dash)
127	DEL	(Delete)
128	Ç	(Majuscule C-Cedilla)
129	ü	(“U” With Diaeresis)
130	é	(“E” With Acute Accent)
131	â	(“A” With Circumflex Accent)
132	ä	(“A” With Diaeresis)
133	à	(“A” With Grave Accent)
134	å	(“A” With A Ring)
135	ç	(Minuscule C-Cedilla)
136	ê	(“E” With Circumflex Accent)
137	ë	(“E” With Diaeresis)
138	è	(“E” With Grave Accent)
139	ï	(“I” With Diaeresis)
140	î	(“I” With Circumflex Accent)
141	ì	(“I” With Grave Accent)
142	Ä	(“A” With Diaeresis)
143	Å	(“A” With A Ring)
144	É	(“E” With Acute Accent)
145	æ	(Latin Diphthong “Ae”)
146	Æ	(Latin Diphthong “AE”)
147	ô	(“O” With Circumflex Accent)

Indeks Karakter	Symbol	Description
148	ö	(“O” With Diaeresis)
149	ò	(“O” With Grave Accent)
150	û	(“U” With Circumflex Accent)
151	ù	(“U” With Grave Accent)
152	ÿ	(“Y” With Diaeresis)
153	Ö	(“O” With Diaeresis)
154	Ü	(“U” With Diaeresis)
155	ø	(Slashed Zero)
156	£	(Pound Sign)
157	Ø	(Slashed Zero)
158	×	(Multiplication Sign)
159	ƒ	(Function Sign)
160	á	(“A” With Acute Accent)
161	Í	(“I” With Acute Accent)
162	ó	(“O” With Acute Accent)
163	ú	(“U” With Acute Accent)
164	ñ	(Lowercase “N” With Tilde)
165	Ñ	(Capital “N” With Tilde)
166	ª	(Feminime Ordinal Indicator)
167	º	(Masculine Ordinal Indicator)
168	¿	(Inverted Question Marks)
169	®	(Registered Trademark)
170	¬	(Logical Negation Symbol)
171	½	(One Half)
172	¼	(Quarter Or One Fourth)
173	¡	(Exclamation Marks)
174	«	(Angle Quotes)
175	»	(Angle Quotes)
176	▒	
177	▒	

Indeks Karakter	Symbol	Description
178	▓	
179	│	(Box Drawing Character)
180	┤	(Box Drawing Character)
181	Á	(“A” With Acute Accent)
182	Â	(“A” With Circumflex Accent)
183	À	(“A” With Grave Accent)
184	©	(Copyright Symbol)
185	╣	(Box Drawing Character)
186	║	(Box Drawing Character)
187	╗	(Box Drawing Character)
188	╝	(Box Drawing Character)
189	¢	(Cent Symbol)
190	¥	(YEN And YUAN Sign)
191	┐	(Box Drawing Character)
192	└	(Box Drawing Character)
193	┴	(Box Drawing Character)
194	┬	(Box Drawing Character)
195	├	(Box Drawing Character)
196	─	(Box Drawing Character)
197	┼	(Box Drawing Character)
198	ã	(“A” With Tilde)
199	Ã	(“A” With Tilde
200	╚	(Box Drawing Character)
201	╔	(Box Drawing Character)
202	╩	(Box Drawing Character)
203	╦	(Box Drawing Character)
204	╠	(Box Drawing Character)
205	═	(Box Drawing Character)
206	╬	(Box Drawing Character)
207	¤	(Generic Currency Sign)

Indeks Karakter	Symbol	Description
208	ð	(Lowercase Letter “Eth”)
209	Ð	(Capital Letter “Eth”)
210	Ê	(“E” With Circumflex Accent)
211	Ë	(“E” With Umlaut)
212	È	(“E” With Grave Accent)
213	ı	(Lowercase Dot Less I)
214	Í	(“I” With Acute Accent)
215	î	(“I” With Circumflex Accent)
216	ï	(“I” With Umlaut)
217	┘	(Box Drawing Character)
218	┌	(Box Drawing Character)
219	█	(Block)
220	▄	(Bottom Half Block)
221	¦	(Vertical Broken Bar)
222	ì	(“I” With Grave Accent)
223	▀	(Top Half Block)
224	Ó	(“O” With Acute Accent)
225	ß	(Letter “Eszett” Or Sharp “S”)
226	Ô	(“O” With Circumflex Accent)
227	Ò	(“O” With Grave Accent)
228	õ	(“O” With Tilde)
229	Õ	(“O” With Tilde)
230	µ	(Letter “Mu” Or Micro Sign)
231	þ	(Lowercase Letter “Thom”)
232	Þ	(Uppercase Letter “Thom”)
233	Ú	(“U” With Acute Accent)
234	Û	(“U” With Circumflex Accent)
235	Ù	(“U” With Grave Accent)
236	ý	(“Y” With Acute Accent)
237	Ý	(“Y” With Acute Accent)

Indeks Karakter	Symbol	Description
238	¯	(Macron Symbol)
239	´	(Acute Accent)
240	≡	(Hyphen)
241	±	(Plus-Minus Sign)
242	‗	(Underline Or Underscore)
243	¾	(Three Quarters)
244	¶	(Paragraph Sign Or Pilcrow)
245	§	(Section Sign)
246	÷	(The Division Sign)
247	¸	(Cedilla)
248	°	(Degree Symbol)
249	¨	(Diaeresis)
250	·	(Interpunct Or Space Dot)
251	¹	(Superscript One)
252	³	(Superscript Three)
253	²	(Superscript Two)
254	■	(Black Square)
255	nbsp	(Non-Breaking Space)

RIWAYAT HIDUP

Syahla Aznadillah, lahir di Depok pada tanggal 16 Juli 2000. Memiliki nama panggilan Syahla. Bertempat tinggal di Jalan Peta Barat No. 06 RT 002/RW 007 Kelurahan Pegadungan, Kecamatan Kalideres, Jakarta Barat. Merupakan anak pertama dari 2 bersaudara dari Bapak Munadih dan Ibu Sumiati.

Pendidikan yang pernah ditempuh yaitu TKQ AR Ridho. Kemudian melanjutkan di SD Negeri 09 Pegadungan dan lulus pada tahun 2012. Menempuh Pendidikan di SMP Negeri 169 Jakarta lulus pada tahun 2015. Pada tahun yang sama melanjutkan pendidikan di SMA Negeri 95 Jakarta dan lulus pada tahun 2018. Tahun 2018 melanjutkan studi di Universitas Islam Negeri Maulana Malik Ibrahim Malang menempuh Program Studi Matematika. Aktif dalam kegiatan akademis dan organisasi di luar kampus seperti HMJ “Integral” Matematika, Lintang Kahuripan, dan KAMAJAYA (Keluarga Mahasiswa Jabodetabek Raya).

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 : Syahla Aznadillah
NIM : 18610054
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi :Modifikasi *Affine Cipher* Menggunakan Algoritma Blum-Blum Shub Dalam Mengamankan Pesan
Pembimbing I : Evawati Alisah, M.Pd
Pembimbing II : Erna Herawati, M.Pd

No	Tanggal	Hal	Tanda Tangan
1.	7 Februari 2022	Konsultasi Bab 1	1.
2.	14 Februari 2022	Revisi Bab 1	2.
3.	17 Februari 2022	Konsultasi Kajian Keagamaan	3.
4.	21 Februari 2022	Konsultasi Bab 2	4.
5.	2 Maret 2022	Konsultasi Bab 3	5.
6.	11 Maret 2022	Konsultasi Kajian Keagamaan	6.
7.	11 Maret 2022	Revisi Bab 3	7.
8.	23 Maret 2022	Acc Bab 1, 2, 3 Oleh Dosen Pembimbing I	8.
9.	25 Maret 2022	Acc Bab 1, 2, 3 Oleh Dosen Pembimbing II	9.
10.	15 April 2022	Konsultasi Revisi Seminar Proposal	10.
11.	23 April 2022	Konsultasi Bab 4	11.
12.	16 Mei 2022	Revisi Bab 4	12.
13.	19 Mei 2022	Konsultasi Bab 4,5	13.
14.	23 Mei 2022	Revisi Bab 5	14.
15.	30 Mei 2022	Acc Bab 4, 5 Oleh Dosen Pembimbing I	15.
16.	30 Mei 2022	Acc Bab 4, 5 Oleh Dosen Pembimbing II	16.
17.	10 Juni 2022	Konsultasi Revisi Seminar Hasil	17.
18.	22 Juni 2022	Acc Keseluruhan	18.

Malang, 22 Juni 2022
Mengetahui,
Ketua Program Studi Matematika

Dr. Elly Susanti, M.Sc
NIP.197411292000122005