BAB II
TINJAUAN PUSTAKA

Pada proses penelitian ini, dimulai dengan melakukan studi kepustakaan terlebih dahulu dari beberapa penelitian yang sudah ada sebelumnya, diantaranya jurnal internasional serta buku yang terkait. Kegiatan ini dilakukan sebagai referensi media sebagai landasan teori. Setelah melakukan studi kepustakaan tersebut, ditemukan beberapa penelitian yang dapat mendorong untuk mengambil topik. Penelitian ini membahas tentang beberapa teknik algoritma yang terkait dengan tema yang diangkat.

2.1 Tinjauan Studi

Kriptografi dapat dikombinasikan dengan metode-metode terkait lainnya sehingga dapat memperkuat kekuatan algoritma kriptografi tersebut. Karena kriptografi memerlukan kunci yang acak sebagai salah satu syarat menjadikan algoritma yang kuat. Sehingga terdapat penelitian dilakukan dengan menggabungkan teknik-teknik algoritma kriptografi yang ada dengan menggunakan algoritma *One Time Pad*, dan pembangkit kunci berupa bilangan acak semu, algoritma *Blum Blum Shub* . Di bawah ini adalah penelitian-penelitian yang berhubungan dengan penelitian skripsi ini, antara lain :

Pada penelitian yang dilakukan M. U. Bokhari, Shadab Alam, dan Faheem Syeed Masoodi [5] terdapat serangkaian serangan *crypanalysis* dengan beberapa skenario, dan menerangkan bahwa dalam sebuah serangan *Exhaustive Key Search* atau serangan *brute force*, kripanalis mencoba seluruh kunci yang mungkin untuk melakukan dekripsi sebuah *ciphertext* dan bisa digunakan untuk beberapa algoritma kriptografi termasuk *stream cipher* kecuali *proveable secure cipher*, meski tidak dalam kondisi layak sekalipun seperti *One Time Pad*.

Suara juga dapat di implementasikan dengan OTP menggunakan MATLAB oleh Y. Saleem, M. Amjad, M. H. Rahman, F. Hayat, T. Izhar, M. Saleem dalam penelitiannya pada tahun 2013 [7]. Mereka menyatakan *One Time Pad* salah satu teknik enkripsi yang dikenal tidak mungkin dipecahkan bila sesuai prosedurnya.

Algoritma *Blum Blum Shub* juga digunakan Praneeth Kumar G dan Vishnu Murthy G [8]. Mereka mengusulkan algoritma kriptografi baru dan menyebutkan bahwa algoritma BBS digunakan sebagai pembangkit kunci dan fungsi genetik "CROSSOVER" dan "MUTATION" pada proses enkripsi dan dekripsi.

S.G.Srikantaswamy dan Dr.H.D.Phaneendra menyatakan bahwa algoritma One Time Pad bisa disebut algoritma sebagai tanpa syarat aman, jika kunci (pad) yang dipakai benar-benar acak di alam [9]. Mereka menunjukkan bahwa satu kali pad dapat digunakan sebagai skema enkripsi efisien dengan melibatkan operasi aritmatika dan logika. Keduanya mengusulkan teknik pembangkit kunci baru, untuk menghasilkan kunci dari setiap panjang hanya dengan memberikan nilai benih, untuk mengenkripsi pesan. Masalah dalam menghasilkan kunci telah diselesaikan dengan menggunakan algoritma pembangkitan kunci tersebut.

Algoritma *Blum Blum Shub* menurut Divyanjali, Ankur, dan Vikas Pareek dianggap aman seperti *quadratic residue problem* [10]. Dengan kata lain, untuk memecahkan BBS setara dengan memecahkan *quadratic residue problem*, yang pada gilirannya, akan memecahkan masalah NP-lengkap yang merupakan basis kriptografi. Karena alasan ini, BBS adalah jenis algoritma yang paling disukai untuk tujuan kriptografi seperti pembangkitan kunci.

Tabel 1. *State of the art*

No	Peneliti	Tahun	Judul	Metode	Masalah	Hasil
1	M.U. Bokhari, Shadab Alam, dan Faheem Syeed Masoodi	2012	Cryptanalysis Techniques for Stream Cipher: A Survey	One Time Pad	Menguji One Time Pad dengan berbagai serangan pada algoritma *Stream Cipher*	One Time Pad termasuk algoritma *cipher* yang terbukti aman oleh serangan
2	Y. Saleem, M. Amjad, M. H. Rahman, F. Hayat, T. Izhar, dan M. Saleem	2013	Speech Encryption Implemetation Of 'One Time Pad Algorithm' In MATLAB	One Time Pad	Diperlukan enkripsi pada suara di lingkungan jaringan yang tidak aman	One Time Pad salah satu teknik enkripsi yang dikenal tidak mungkin dipecahkan bila sesuai prosedurnya. Implemetasi-nya pada enkripsi suara di MATLAB
3	Praneeth Kumar G, dan Vishnu Murthy G	2010	Design of a Novel Cryptographic Algorithm using Genetic Functions	Blum Blum Shub	Keamanan informasi merupakan kunci dari komputasi modern	Blum Blum Shub sebagai pembangkit kunci dan fungsi genetik pada proses

						enkripsi dan dekripsi di algoritma kriptografi
4	S.G.Srikantaswamy, dan Dr.H.D.Phaneendra	2011	Enhanced OneTime Pad Cipher with More Arithmetic and Logical Operations with Flexible Key Generation Algorithm	One Time Pad	One Time Pad memerlukan pembangkit kunci yang fleksibel	One Time Pad akan tidak bisa dipecahkan bila kuncinya benar-benar acak, maka diajukan pembangkit kunci acak baru
5	Divyanjali, Ankur, dan Vikas Pareek	2014	An Overview of Cryptographically Secure Pseudorandom Number generators and BBS	Blum Blum Shub	Diperlukan survei literatur kriptografi generator nomor acak semu yang aman	Algoritma BBS dianggap aman seperti *quadratic residue problem*. Dengan kata lain, untuk memecahkan BBS setara dengan memecahkan *quadratic residue problem*.

Dari tinjauan studi tersebut, akan digunakan teknik Blum Blum Shub sebagai pembangkit kunci acak semu pada algoritma *One Time Pad*. Pada penggunaannya nanti, teknik OTP sebagai *stream cipher* akan divariasi pada bagian kunci, dan akan menghasilkan *cipherteks* berupa bilangan hexadesimal untuk memperkuat pesan rahasia, karena tidak hanya mengandalkan bilangan desimal dan biner saja seperti pada algoritma One Time Pad tradisional.

2.1.1 Aspek Ancaman Keamanan

Peranan sistem komputer sebagai penyedia informasi dapat dimanfaatkan sebagai landasan untuk menentukan jenis ancaman terhadap suatu sistem komputer, sehingga diperoleh jenis-jenis ancaman terhadap sistem komputer yang dapat dikelompokan menjadi empat yaitu [11]:

a. *Interruption*, adalah suatu ancaman terhadap *availability*, informasi atau data yang ada dalam sistem komputer dirusak, dihapus, sehingga jika diperlukan maka sudah tidak ada lagi.
b. *Interception*, adalah ancaman terhadap kerahasiaan (*secrecy*). Informasi yang ada di dalam sistem disadap oleh pihak yang tidak berwenang.
c. *Modification*, adalah ancaman terhadap keutuhan (integritas). Pihak yang tidak berwenang sukses menyadap hilir mudik informasi yang sedang dikirim, kemudian mengubahnya sesuai dengan kehendak pihak tersebut.
d. *Fabrication*, adalah ancaman terhadap keutuhan (integritas). Pihak yang tidak berwenang sukses meniru atau memalsukan suatu informasi sehingga pihak yang memperoleh informasi tersebut berpikir informasi tersebut bermula dari pihak yang dikehendaki oleh si penerima informasi tersebut.

2.1.2 Aspek Keamanan

Kriptografi bukan hanya memfasilitasi kerahasiaan dalam telekomunikasi, namun juga meliputi aspek keamanan komputer meliputi komponen-komponen berikut ini [1, 11]:

i. *Authentication*. Penerima pesan mampu meyakini keaslian pengirimnya. Penyerang tidak dapat berpura-pura tehadap orang lain.
ii. *Integrity*. Penerima wajib memeriksa apakah pesan telah diubah saat dalam perjalanan atau tidak. Pihak penyusup semestinya tidak dapat memasukkan tambahan ke dalam pesan, mengurangi atau mengubah pesan selama data berada di perjalanan.
iii. *Nonrepudiation*. Pengirim semestinya tidak mampu menyangkal bahwa dialah pengirim pesan yang sebenarnya. Tanpa kriptografi, seseorang dapat menyangkal bahwa dialah pengirim email yang sebenarnya.
iv. *Authority*. Informasi yang ada dalam sistem jaringan semestinya hanya dapat diubah oleh pihak yang berwenang. Modifikasi yang tidak dikehendaki, dapat berupa penulisan tambahan pesan, pengubahan isi, pengubahan status, penghapusan, pembuatan pesan baru (pemalsuan), atau menyalin pesan untuk digunakan kemudian oleh penyerang.
v. *Confidentiality*. Adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Kerahasiaan ini umumnya berkaitan dengan informasi yang diberikan ke pihak lain.
vi. *Privacy*. Lebih ke arah data - data yang bersifat pribadi.
vii. *Availability*. Aspek availabilitas berhubungan dengan ketersediaan informasi ketika diperlukan. Sistem informasi yang diserang atau dijebol dapat menghambat atau mengabaikan akses ke informasi.
viii. *Acces Control*. Aspek ini berkaitan dengan prosedur pengaturan akses ke informasi. Hal ini umumnya berkaitan dengan persoalan *otentifikasi* dan *privasi*. Pengendalian akses sering dilakukan dengan

menggunakan kombinasi user id dan password ataupun dengan cara kerja lain.

2.1.3 Definisi dan Terminologi Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani : "*cryptos*" yang artinya "*secret*" (rahasia), sedangkan "*graphein*" yang artinya "*writing*" (tulisan). Jadi, kriptografi berarti "*secret writing*" (tulisan rahasia). Ada beberapa definisi kriptografi yang telah disampaikan di dalam beberapa literatur. Kriptografi adalah ilmu dan seni untuk mengawal keamanan pesan. *(Cryptography is the art and science of keeping messages secure).* Selain itu ada definisi lain bahwa kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi.

Pengertian tentang kriptografi yang disampaikan dalam buku-buku lama (sebelum tahun 1980-an) bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasaiaan pesan dengan cara menyandikan ke dalam wujud yang tidak dapat dipahami lagi maknanya. Definisi ini mungkin sesuai pada masa lalu di mana kriptografi dipakai untuk keamanan komunikasi penting seperti komunikasi dikalangan militer, diplomat, dan mata-mata. Namun saat ini kriptografi lebih dari sekedar *privacy*, melainkan juga untuk tujuan data *integrity*, *authentication*, dan *non-repudiation.*

Kata "seni" di dalam definisi di atas berasal dari kenyataan sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai prosedur yang unik untuk merahasiakan pesan. Prosedur unik tersebut mungkin berbeda-beda pada setiap pelaku kriptografi sehingga setiap cara menulis pesan rahasia, pesan mempunyai nilai estetika tersendiri sehingga kriptografi berkembang menjadi sebuah seni merahasiakan pesan (kata "*graphy*" di dalam "*cryptography*" itu sendiri sudah mengandung sebuah seni). Pada perkembangan lebih lanjut, kriptografi berkembang menjadi disiplin ilmu tersendiri karena

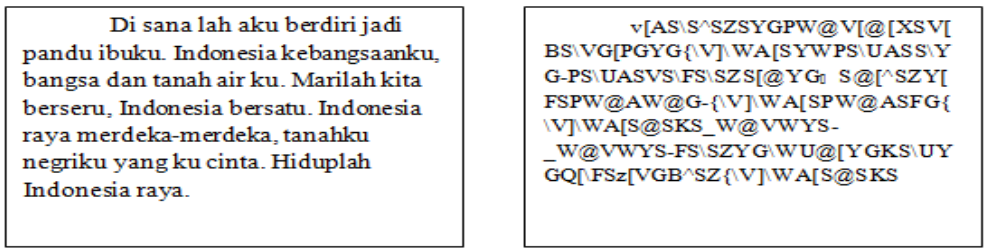
teknik-teknik kriptografi dapat dirumuskan secara matematik sehingga menjadi sebuah prosedur yang formal.

Di dalam kriptografi kita akan sering menjumpai berbagai istilah atau terminologi. Beberapa istilah yang penting untuk diketahui diuraikan di bawah ini:

2.1.3.1 Pesan, Plainteks, dan Cipherteks

Pesan *(message)* adalah data atau informasi yang dapat dibaca dan dimengerti maknanya. Nama lain untuk pesan adalah plainteks *(plaintext)* atau teks jelas *(cleartext)*. Pesan dapat berwujud data atau informasi yang dikirim (saluran telekomunikasi, melalui kurir, dsb) atau yang disimpan di dalam media perekaman (kertas, strorage, dsb). Pesan yang disimpan tidak hanya berupa teks, tetapi juga berbentuk citra (*image*), suara/bunyi (*audio*), dan video, atau berkas biner lainnya.

Agar pesan tidak dapat dimengerti maknanya oleh pihak lain, maka pesan perlu disandikan ke wujud lain yang tidak dapat dipahami. Wujud pesan yang tersandi disebut cipherteks *(ciphertext)* atau kriptogram *(cryptogram)*. Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca. Gambar 1. memperlihatkan contoh plainteks berupa pesan teks, serta cipherteks yang saling berkoresponden. Untuk diperhatikan bahwa plainteks dapat dibaca dengan jelas, tetapi cipherteks sudah tidak dapat lagi dimengerti maknanya. Melalui proses yang berkebalikan, cipherteks dapat ditransformasikan kembali menjadi plainteks asal. Berikut ini adalah contoh gambar plainteks dan cipherteks.



(a) Plainteks (teks) (b) Cipherteks dari (a)

Gambar 1 : Contoh plainteks dan cipherteks

2.1.3.2 Pengirim dan penerima

Komunikasi data melibatkan pertukaran pesan antara dua entitas. Pengirim (*sender*) adalah entitas yang mengirim pesan kepada entitas lainnya. Penerima (*receiver*) adalah entitas yang menerima pesan. Entitas di sini dapat berwujud orang, mesin (komputer), kartu kredit, dan sebagainya. Jadi, orang dapat bertukar pesan dengan orang lainnya (contoh : Freddy berkomunikasi dengan Rocky), sedangkan di dalam jaringan komputer mesin (komputer) berkomunikasi dengan mesin (contoh : mesin ATM berkomunikasi dengan komputer server di bank).

Pengirim tentu menginginkan pesan dapat dikirim secara aman, yaitu ia yakin bahwa pihak lain tidak dapat membaca isi pesan yang ia kirim. Solusinya adalah dengan cara menyandikan pesan menjadi cipherteks.

2.1.3.3 Enkripsi dan dekripsi

Proses menyandikan plainteks menjadi cipherteks disebut enkripsi (*encryption*) atau *enciphering*. Sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi (*decryption*) atau *deciphering*. Enkripsi dan dekripsi dapat diterapkan baik pada pesan yang dikirim maupun pada pesan tersimpan.

2.1.3.4 Cipher dan kunci

Algoritma kriptografi disebut juga *cipher* yaitu aturan untuk *enciphering* dan *deciphering*, atau fungsi matematika yang dipakai untuk enkripsi dan

dekripsi. Beberapa *cipher* memerlukan algoritma yang berbeda untuk *enciphering* dan *deciphering*.

Konsep matematis yang mendasari algoritma kriptografi adalah relasi antara dua buah himpunan yaitu himpunan yang berisi elemen-elemen plainteks dan himpunan yang berisi cipherteks. Enkripsi dan dekripsi merupakan fungsi yang memetakan elemen-elemen antara kedua himpunan tersebut. Misalkan P menyatakan plainteks dan C menyatakan cipherteks, maka fungsi enkripsi E memetakan P ke C,

$$E(P) = C \qquad (1)$$

Dan persamaan dekripsi D memetakan C ke P,

$$D(C) = P \qquad (2)$$

Karena proses enkripsi kemudian dekripsi mengembalikan pesan ke pesan asal, maka kesamaan tersebut harus benar,

$$D(E(P)) = P \qquad (3)$$

Keamanan algoritma kriptografi sering diukur dari banyaknya kerja (*work*) yang diperlukan untuk memecahkan cipherteks menjadi plainteksnya tanpa mengetahui kunci yang dipakai. Kerja ini dapat dibandingkan dengan waktu, memori, uang, dan lain-lain. Semakin besar usaha yang diperlukan, yang berarti juga semakin lama waktu yang dibutuhkan, maka semakin kuat algoritma kriptografi tersebut, yang berarti semakin aman dipakai untuk menyandikan pesan.

Jika keamanan kriptografi ditentukan dengan menjaga kerahasiaan algoritmanya, maka algoritma kriptografi tersebut dipanggil algoritma *restricted*. Algoritma *restricted* memiliki sejarah tersendiri di dalam kriptografi. Algoritma *restricted* umumnya dipakai oleh sekelompok pihak untuk bertukar pesan satu sama lain. Mereka membuat suatu algoritma enkripsi dan algoritma enkripsi

tersebut hanya diketahui oleh anggota kelompok itu saja. Tetapi, algoritma *restricted* tidak sesuai untuk saat ini, karena setiap ada anggota kelompok keluar, maka algoritma kriptografi harus diganti.

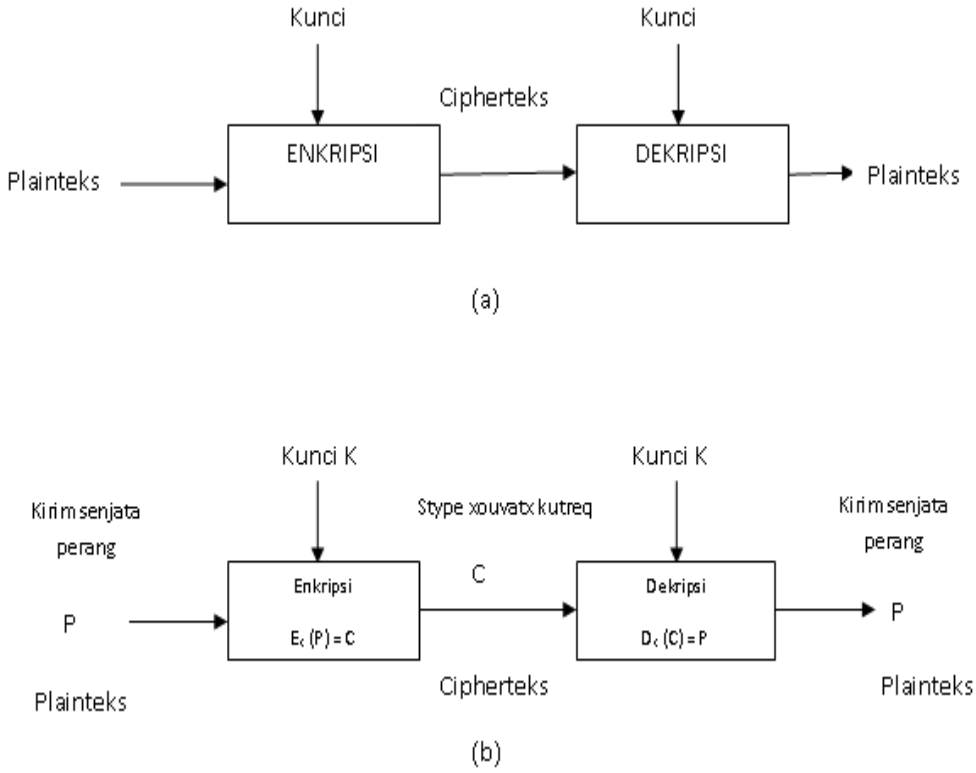
Kriptografi modern menanggulangi persoalan di atas dengan pemakaian kunci, yang dalam hal ini algoritma tidak lagi dirahasiakan, tetapi kunci harus diawasi kerahasiaannya. Kunci (*key*) adalah parameter yang dipakai untuk transformasi *encriphering* dan *deciphering*. Kunci umumnya berwujud string atau deretan bilangan. Dengan memakai kunci K, maka peranan enkripsi dan dekripsi dapat ditulis sebagai

$$E_K(P) = C \text{ dan } D_K(C) = P \qquad (4)$$

Dan kedua kunci ini memenuhi

$$D_K(E_K(P)) = P \qquad (5)$$

Gambar 2 (a). menunjukkan skema enkripsi dan dekripsi dengan memakai kunci, sedangkan Gambar 2 (b). menggambarkan enkripsi dan dekripsi terhadap sebuah pesan.



Gambar 2 : Skema enkripsi dan dekripsi

Istilah "*cipher*" sering disamakan dengan kode (*code*). Kode memiliki riwayat tersendiri di dalam kriptografi. Sebetulrnya kedua istilah ini tidak sama maknanya. Jika cipher atau sandi adalah transformasi karakter-karakter atau bit-bit tanpa meperhatikan susunan bahasa pesan, maka kode sering diacu sebagai prosedur yang mengganti setiap plainteks dengan kata kode, misalnya :

pasukan musuh datang dikodekan menjadi *kelinci lemah tidur*

Kode juga dapat berupa deretan angka dan huruf yang tidak bermakna, seperti :

pasukan musuh datang dikodekan menjadi *fhgurt mosjqt kcsfyl*

Transformasi dari plainteks menjadi kode sering disebut *encoding*, sedangkan transformasi kebalikannya sering disebut *decoding*.

2.1.3.5 Sistem Kriptografi

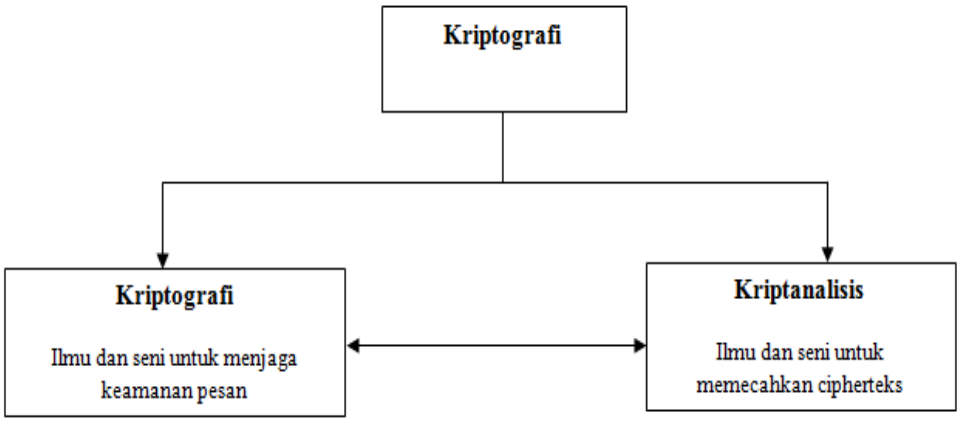
Kriptografi menyusun sebuah sistem yang dinamakan seistem kriptografi. Sistem kriptografi (*cryptosystem*) adalah kumpulan yang terdiri dari algoritma kriptografi, semua plainteks dan cipherteks yang mungkin, dan kunci. Di dalam sistem kriptografi, *cipher* hanyalah salah satu komponen saja.

2.1.3.6 Penyadap

Penyadap (*eavesdropper*) adalah orang yang mencoba menangkap pesan selama ditransmisikan. Tujuan penyadap adalah untuk mendapatkan informasi sebanyak-banyaknya mengenai sistem kriptografi yang dipakai untuk berkomunikasi dengan maksud untuk memecahkan cipherteks. Nama lain penyadap : *enemy, adversary, intruder, interceptor, bad guy*. Ron Riverst, seorang pakar kriptografi, menyatakan bahwa *cryptography is about communication in the presence of adversaries* (Kriptografi adalah perihal berkomunikasi dengan keberadaan pihak musuh).

2.1.3.7 Kriptanalisis dan kriptologi

Kriptografi berkembang sedemikian rupa sehingga melahirkan bidang yang berlawanan yaitu kriptanalisis. Kriptanalisis *(cryptanalysis)* adalah ilmu dan seni untuk memecahkan cipherteks menjadi plainteks tanpa mengetahui kunci yang dipakai. Pelakunya disebut kriptanalis. Jika seorang kriptografer *(cryptographer)* mentransformasikan plainteks menjadi cipherteks dengan suatu algoritma dan kunci maka sebaliknya seorang kriptanalis berusaha untuk memecahkan cipherteks tersebut untuk menemukan plainteks atau kunci. Kriptologi *(cryptology)* adalah studi mengenai kriptografi dan kriptanalisis. Baik kriptorafi maupun kriptanalisis keduanya saling berkaitan. Gambar 3. memperlihatkan pohon kriptologi.



Gambar 3 : Hubungan antara kriptografi dan kriptanalisis

Sebagian para praktisi sering menggunakan istilah kriptografi dan kriptologi secara bergantian, sebagian lagi membedakan bahwa kriptografi mengecu pada penggunaan praktisi teknik-teknik kriptografi, sedangkan kriptologi mengacu pada subjek sebagai bidang studi (seperti halnya biologi, geologi, antropologi, dan sebagainya).

2.1.4 *One-Time Pad* dan *Cipher* Aliran *(Stream Cipher)*

Cipher aliran *(stream cipher)* mengenkripsi teks-asli (plainteks) menjadi teks sandi (cipherteks) bit per bit (1 bit setiap kali transformasi). Pertama kali diperkenalkan oleh Vernam melalui algoritma yang dikenal dengan nama *Vernam cipher*.

Sandi Vernam atau *Vernam cipher* diadopsi dari *one – time-pad cipher*, yang dalam hal ini karakter diganti dengan bit (0 atau 1). Teks tersandi diperoleh dengan melakukan penjumlahan modulo 2 satu bit teks-asli dengan satu bit kunci :

$$C_i = (P_i + K_i) \bmod 2, \qquad (6)$$

yang dalam hal ini :

P_i = bit plainteks

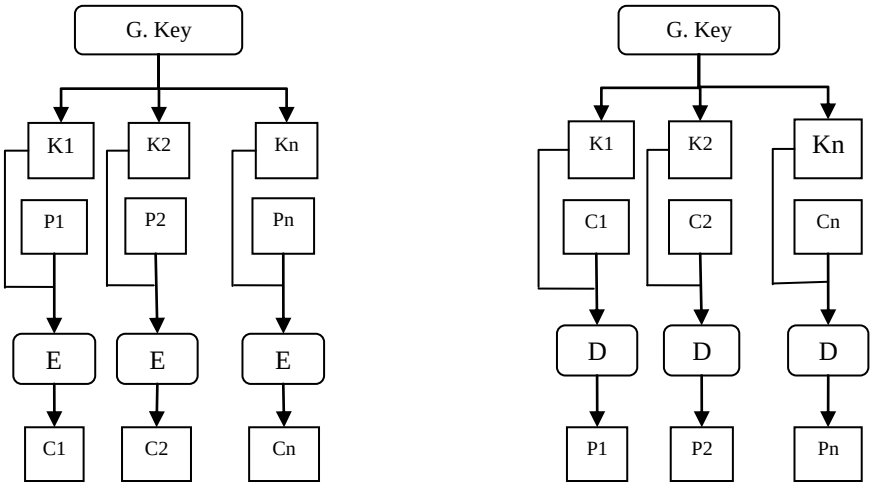
K_i = bit kunci

C_i = bit cipherteks

Plainteks diperoleh dengan melakukan pengurangan modulo 2 satu bit cipherteks dengan satu bit kunci :

$$P_i = (C_i - K_i) \bmod 2 \qquad (7)$$

Oleh karena itu dapat diartikan bahwa *cipher* aliran merupakan versi lain dari *one-time-pad*.

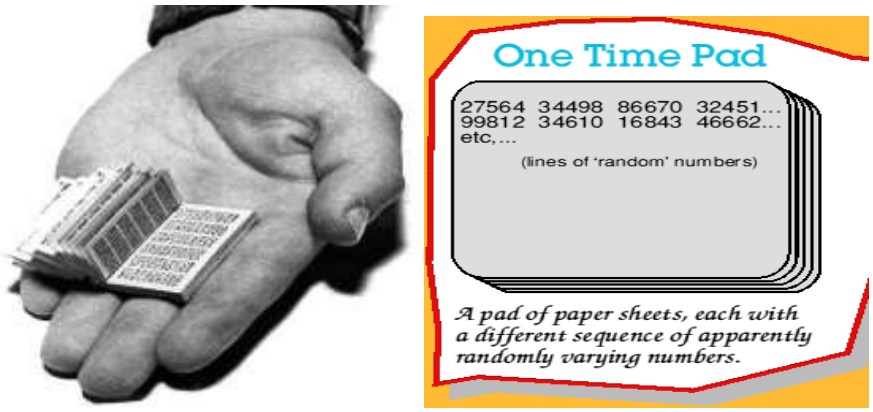


Gambar 4 : Konsep *cipher* aliran

Keamanan sistem *stream cipher* bergantung seluruhnya pada pembangkit aliran kunci. Jika pembangkit mengeluarkan aliran kunci yang seluruhnya nol, maka cipherteks sama dengan plainteks, dan proses enkripsi menjadi tidak ada artinya. Jika pembangkit mengeluarkan aliran kunci yang benar-benar acak *(truly random),* maka algoritma enkripsinya sama dengan *one time pad* dengan tingkat keamanan yang sempurna. Pada kasus ini, aliran kunci sama panjangnya dengan panjang plainteks, dan kita mendapatkan *cipher* aliran sebagai *unbreakable*

cipher. Semakin acak keluaran yang dihasilkan oleh pembangkit aliran kunci, semakin sulit kriptanalisis memecahkan cipherteks. Pada *Stream Cipher* salah satu algoritma yang mendekati kesempurnaan adalah *One Time Pad*.

Cipher yang tidak dapat dipecahkan dikatakan memiliki tingkat kerahasiaan yang sempurna *(perfect secrecy)*. Satu-satunya algoritma kriptografi yang sempurna aman dan tidak dapat dipecahkan adalah one the pad (secara matematis Shannon telah membuktikan bahwa OTP tidak dapat dipecahkan). OTP ditemukan pada tahun 1917 oleh Vernam dan Major Joseph Mauborge. *One Time Pad* (*pad* = kertas bloknot) adalah kertas yang berisi deretan karakter-karakter kunci yang berisi huruf-huruf yang tersusun acak. Shanon membuktikan apabila sandi one time pad diterapkan secara benar maka akan mencapai rahasia sempurna, (Shannon, 1949). Sebuah sandi disebutkan demikian jika pasangan teks asli dan teks sandi tidak memiliki hubungan statistik sehingga sulit bagi penyerang untuk melakukan analisis sandi atau analisis statistik. [12] (Gambar 5) Satu pad hanya digunakan sekali *(one-time)* saja untuk mengenkripsi pesan, setelah itu pad yang telah digunakan dihancurkan supaya tidak dipakai kembali untuk mengenkripsi pesan yang lain.



Gambar 5 : *One Time Pad*

Aturan enkripsi yang digunakan persis sama seperti pada *Vigenere Cipher*. Pengirim pesan menggunakan setiap karakter kunci untuk mengenkripsi satu

karakter plainteks. Enkripsi dapat dinyatakan sebagai penjumlahan modulo 26 dari satu karakter plainteks dengan satu karakter kunci one-time pad :

$$C_i = (P_i + K_i) \bmod 26 \qquad (8)$$

Yang dalam hal ini, P_i adalah plainteks atau huruf ke-i, K_i adalah huruf kunci ke-i, dan C_i adalah huruf cipherteks ke-i. Perhatikan bahwa panjang kunci sama dengan panjang plainteks, sehingga tidak ada kebutuhan mengulang penggunaan kunci selama proses enkripsi.

Setelah pengirim mengenkripsikan pesan dengan kata kunci, ia menghancurkan kunci tersebut (oleh karena itu disebut sekali pakai atau *one-time*). Penerima pesan menggunakan kunci yang sama untuk mengdekripsikan karakter-karakter cipherteks menjadi karakter-karakter plainteks dengan persamaan:

$$P_i = (C_i - K_i) \bmod 26 \qquad (9)$$

Jika karakter yang digunakan adalah anggota himpunan 256 karakter (seperti karakter dengan pengkodean ASCII), maka enkripsi dapat dinyatakan sebagai penjumlahan modulo 256 dari satu karakter plainteks dengan satu karakter kunci one-time pad [13]:

$$Ci = (Pi + Ki) \bmod 256 \qquad (10)$$

Penerima pesan menggunakan pad yang sama untuk mendekripsikan cipherteks menjadi plainteks dengan persamaan:

$$Pi = (Ci - Ki) \bmod 256 \qquad (11)$$

Sebagai contoh, proses enkripsi dapat dilihat dibawah :

Plainteks: ONETIMEPAD

Kunci: TBFRGFARFM

Misalkan A = 0, B = 1, C = 2, D = 3, E = 4, F = 5, …, Y = 24, Z = 25.

Didapat cipherteks: HOJKOREGFP

yang mana diperoleh sebagai berikut:

(O + T) mod 26 = (14 + 19) mod 26 = 7 = H

(N + B) mod 26 = (13 + 1) mod 26 = 14 = O

(E + F) mod 26 = (4 + 5) mod 26 = 9 = J

(T + R) mod 26 = (19 + 17) mod 26 = 10 = K

(I + G) mod 26 = (8 + 6) mod 26 = 14 = O

(M + F) mod 26 = (12 + 5) mod 26 = 17 = R

(E + A) mod 26 = (4 + 0) mod 26 = 4 = E

(P + R) mod 26 = (15 + 17) mod 26 = 6 = G

(A + F) mod 26 = (0 + 5) mod 26 = 5 = F

(D + M) mod 26 = (3 + 12) mod 26 = 15 = P

Meskipun OTP merupakan *cipher* yang sempurna aman, namun faktanya ia tidak digunakan secara universal dalam aplikasi kriptografi sebagai satu-satunya sistem *cipher* yang tidak dapat dipecahkan (hanya sedikit sistem komunikasi yang menggunakan OTP). Malahan orang masih tetap menggunakan sistem *cipher* yang dapat dipecahkan, sebagaimana dikatakan oleh seorang praktisi kriptografi dibawah ini :

"As a practical person, I've observed that one-time pads are theoretically unbreakable, but practially very weak. By contrast, conventional cipher are theoretically breakable, but practically strong" – Steve Bellovin

Alasan mengapa *One Time Pad* jarang digunakan adalah dari segi kepraktisan, yaitu :

a. Karena panjang kunci harus sama dengan panjang pesan, maka OTP hanya cocok untuk pesan berukuran kecil. Semakin besar ukuran pesan, semakin besar pula ukuran kunci. Pada aplikasi kriptografi untuk mengenkripsi data tersimpan, timbul masalah dalam penyimpanan kunci. Pada aplikasi kriptografi untuk komunikasi pesan, timbul masalah dalam pendistribusian kunci.
b. Karena kunci dipakai secara acak, maka 'tidak mungkin' pengirim dan penerima membangkitkan kunci yang sama secara simultan. Jadi, salah seorang dari mereka harus membangkitkan kunci lalu megirimkannya ke pihak lain.

Karena kerahasiaan kunci harus terjamin, maka perlu adanya perlindungan selama pengiriman kunci. Jika hanya ada satu saluran komunikasi, maka pengirim dan penerima pesan perlu barisan kunci lain untuk melindungi kunci pertama, kunci ketiga untuk melindungi kunci kedua dan seterusnya. Hal ini menghasilkan barisan kunci yang tidak berhingga banyaknya. Mengirimkan barisan kunci melalui saluran komunikasi yang digunakan untuk pengiriman pesan juga tidak praktis karena pertimbangan lalu lintas *(traffic)* pesan yang padat. Oleh karena itu, OTP hanya dapat digunakan jika tersedia saluran komunikasi kedua yang cukup aman untuk mengirim kunci. Saluran kedua ini umumnya lambat dan mahal. Misalnya pada perang dingin antara Amerika Serikat dan Uni Soviet dahulu, kunci dibangkitkan lalu disimpan, lalu dikirimkan melalui jasa kurir yang aman. Penting diingat bahwa saluran kedua yang aman tersebut umumnya lambat dan mahal.

2.1.5 Algoritma yang Aman

Unbreakable cipher merupakan klaim yang dibuat oleh kriptografer terhadap algoritma kriptografi yang dirancangnya. Kriptografer sering menyatakan bahwa *cipher* yang dirancangnya tidak dapat dipecahkan.

Pada kriptografi terdapat algoritma yang tidak dapat dipecahkan dan untuk merancang algoritma *unbreakable cipher*, ada dua syarat yang harus dipenuhi :

a. Kunci yang dipilih harus secara acak (yaitu, setiap kunci harus mempunyai peluang yang sama untuk terpilih).
b. Panjang kunci harus sama dengan panjang plainteks yang akan dienkripsikan.

Kedua syarat tersebut dapat menyebabkan plainteks sama belum tentu dienkripsi menjadi cipherteks yang sama. Itu artinya kriptanalisis akan mendapatkan hasil bahwa setiap cipherteks yang didekripsikannya mungkin mengahasilkan beberapa plainteks bermakna. Hal ini akan membingungkannya dalam menentukan plainteks mana yang benar [3].

Suatu algoritma dikatakan aman, jika tidak ditemukan plainteksnya walaupun berapa banyaknya cipherteks yang dimiliki kriptanalis. Seluruh algoritma dapat dipecahkan dengan mencoba satu per satu seluruh kemungkinan kunci dan memeriksa apakah plainteks sesuai *(brute force attack).* Kriptografi dikatakan aman jika memiliki keadaan sebagai berikut [1]:

a. Jika harga untuk memecahkan algoritma lebih besar daripada nilai informasi yang dibuka.
b. Bila waktu yang diperlukan untuk memecahkan algoritma tersebut lebih lama daripada lama waktu yang diperlukan oleh informasi.
c. Jika jumlah data yang dienkrip dengan kunci algoritma yang sama lebih sedikit dari jumlah data yang diperlukan untuk memecah algoritma tersebut.

Hal-hal yang dibutuhkan untuk memecahkan suatu algoritma kriptografi adalah [14]:

a. Mendapatkan cipherteks sebanyak mungkin, baik dari jaringan internet, telepon, baik dari saluran telekomunikasi lainnya. Semakin banyak cipherteks yang didapat semakin mudah dalam pemecahan.
b. Mencari tahu bahasa yang dipakai untuk suatu pesan.
c. Algoritma yang dipakai.

d. Mencari informasi tentang pengirim dan penerima. Apabila menyangkut perbedaan warga negara, kemungkinan besar memakai bahasa Inggris. Dan bila latar belakang penerima atau direktur, kemungkinan nama-nama tersebut mudah dikenali sehingga dapat membantu memecahkan suatu pesan tertulis yang sudah menjadi cipherteks.
e. Membuat alat bantu seperti tabel vigenere atau membuat tabel-tabel pola untuk dapat melihat pola-pola huruf apa saja yang sering muncul pada cipherteks tersebut.
f. Mendapatkan kunci dan plainteksnya. Hal tersebut dapat dilakukan tanpa menguasai ilmu kriptografi sedikit pun. Contohnya adalah berpura-pura menjadi atasan administrator (di luar sistem kriptografi).

2.1.6 Algoritma Blum Blum Shub

Algortima Blum Blum Shub adalah algoritma pembangkit bilangan acak semu yang paling sederhana dan paling mangkus secara kompleksitas teoritis. *BBS* dibuat pada tahun 1986 oleh Lenore Blum, Manuel Blum, dan Michael Shub yang dirancang dengan dasar teori bilangan. Untuk membangkitkan bilangan acak dengan *BBS*, algoritmanya adalah sebagai berikut [3] :

1. Pilih dua bilangan prima rahasia, *p* dan q, yang masing-masing kongruen dengan 3 mod 4 (dalam prakteknya bilangan prima yang digunakan adalah bilangan besar) dengan persamaan :

   $p \equiv 3 \bmod 4$ *dan* $q \equiv 3 \bmod 4$ (12)
2. Dari persamaan (12), kalikan *p* dan *q* untuk mencari n yang disebut bilangan bulat Blum dengan persamaan :

   $n = p.q$ (13)
3. Pilih bilangan bulat acak lain, *s*, sebagai umpan sedemikian sehingga:
   1. $2 \leq s < n$
   2. *s* dan *n* relatif prima

   kemudian hitung x_0 dengan persamaan :

   $x_0 = s^2 \bmod n$ (14)

4. Barisan bilangan acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan :
   i. Hitung x_i dengan persamaan :
      $x_i = x_{i-1}^{\ 2} \bmod n$ (15)
   ii. Hasilkan z_i yang merupakan bit-bit dari x_i. Bit yang diambil bisa merupakan *LSB* (*Least Significant Bit*), bilangan acak tidak harus 1 bit *LSB* tetapi juga bisa *j* buah bit (*j* adalah bilangan bulat positif yang tidak melebihi ($log_2(log_2$ n)).
5. Barisan bit acak adalah $z_1, z_2, z_3, ..., z_i$

Sebagai contoh terhadap algoritma Blum Blum Shub akan dijelaskan sebagai berikut:

1. Pilih dua bilangan prima rahasia, *p* dan q, yang masing-masing kongruen dengan 3 mod 4 seperti di persamaan (12):
   i. $p \equiv 3 \bmod 4$ maka $p \bmod 4 = 3$, sehingga diambil $\boldsymbol{p = 11}$
   ii. $q \equiv 3 \bmod 4$ maka $q \bmod 4 = 3$, sehingga diambil $\boldsymbol{q = 19}$
2. Kemudian dari persamaan (12), mencari nilai n dengan mengkalikan *p* dan *q*:
   $n = p.q$
   $n = 11.19$
   $\boldsymbol{n = 209}$
3. Lalu pilih bilangan bulat acak lain, *s*, sebagai umpan sedemikian sehingga:
   i. $2 \leq s < n$
   ii. *s* dan *n* relatif prima

   Relatif prima adalah hasil pemfaktoran dari FPB (Faktor Persekutuan Terbesar) dari 2 bilangan, hanya terdapat satu angka hasil faktor yang sama.

   n = 209, dengan FPB = **1**, 11, 19, 209
   s = 9, dengan FPB = **1**, 3, 9

Didapat s = 9 dan relatif prima dengan n = 209, kemudian hitung x_0 dengan persamaan (14):

$x_0 = s^2 \bmod n$

$x_0 = 9^2 \bmod 209$

$x_0 = 81 \bmod 209$

$x_0 = 81$

4. Barisan bilangan acak dihasilkan dengan melakukan iterasi berikut sepanjang yang diinginkan :
   i. Hitung x_i dengan persamaan (15):

$x_1 = x_0^{\,2} \bmod n = 81^2 \bmod 209 = 6561 \bmod 209 = 82$

$x_2 = x_1^{\,2} \bmod n = 82^2 \bmod 209 = 6724 \bmod 209 = 36$

$x_3 = x_2^{\,2} \bmod n = 36^2 \bmod 209 = 1296 \bmod 209 = 42$

$x_4 = x_3^{\,2} \bmod n = 42^2 \bmod 209 = 1764 \bmod 209 = 92$

$x_5 = x_4^{\,2} \bmod n = 92^2 \bmod 209 = 8464 \bmod 209 = 104$

$x_6 = x_5^{\,2} \bmod n = 104^2 \bmod 209 = 10816 \bmod 209 = 157$

$x_7 = x_6^{\,2} \bmod n = 157^2 \bmod 209 = 24649 \bmod 209 = 196$

$x_8 = x_7^{\,2} \bmod n = 196^2 \bmod 209 = 38416 \bmod 209 = 169$

$x_9 = x_8^{\,2} \bmod n = 169^2 \bmod 209 = 28561 \bmod 209 = 137$

$x_{10} = x_9^{\,2} \bmod n = 137^2 \bmod 209 = 18769 \bmod 209 = 168$

$x_{11} = x_{10}^{\,2} \bmod n = 168^2 \bmod 209 = 28224 \bmod 209 = 9$

$x_{12} = x_{11}^{\,2} \bmod n = 9^2 \bmod 209 = 81 \bmod 209 = 81$

$x_{13} = x_{12}^{\,2} \bmod n = 81^2 \bmod 209 = 65651 \bmod 209 = 82$

$x_{14} = x_{13}^{\,2} \bmod n = 82^2 \bmod 209 = 6724 \bmod 209 = 36$

$x_{15} = x_{14}^{\,2} \bmod n = 36^2 \bmod 209 = 1296 \bmod 209 = 42$

$x_{16} = x_{15}^{\,2} \bmod n = 42^2 \bmod 209 = 1764 \bmod 209 = 92$

. .

. .

. .

Dan seterusnya tergantung banyak karakter, karena 1 karakter = 8bit.

ii. Hasilkan z_i yang merupakan bit-bit dari x_i. Bit yang diambil bisa merupakan *LSB* (*Least Significant Bit*).

x_1 = 82 ; z_1= 0 (karena genap)

x_2 = 36; z_2= 0 (karena genap)

x_3 = 42; z_3= 0 (karena genap)

x_4 = 92; z_4= 0 (karena genap)

x_5 = 104; z_5= 0 (karena genap)

x_6 = 157; z_6= 1 (karena ganjil)

x_7 = 196; z_7= 0 (karena genap)

x_8 = 169; z_8= 1 (karena ganjil)

x_9 = 137; z_9= 1 (karena ganjil)

x_{10} = 168; z_{10}= 0 (karena genap)

x_{11} = 9; z_{11}= 1 (karena ganjil)

x_{12} = 81; z_{12}= 1 (karena ganjil)

x_{13} = 82; z_{13}= 0 (karena genap)

x_{14} = 36; z_{14}= 0 (karena genap)

x_{15}= 42; z_{15}= 0 (karena genap)

x_{16} = 92; z_{16}= 0 (karena genap)

. .

. .

Bilangan acak tidak harus 1 bit *LSB* tetapi juga bisa *j* buah bit (*j* adalah bilangan bulat positif yang tidak melebihi ($log_2(log_2$ n)).

$n = p.q$

n = 11.19

n = 209

j = 2 *(j tidak melebihi* $(log_2(log_2\ 209))$ *= 2,94623)*

x_1 = 82 = 1010010 ; $z_1 = 82 \equiv 2 (\text{mod } 2^2) = 10_{\text{basis 2}}$

x_2 = 36 = 100100 ; $z_2 = 36 \equiv 0 (\text{mod } 2^2) = 00_{\text{basis 2}}$

x_3 = 42 = 101010 ; $z_3 = 42 \equiv 2 (\text{mod } 2^2) = 10_{\text{basis 2}}$

x_4 = 92 = 1011100 ; $z_4 = 92 \equiv 0 (\text{mod } 2^2) = 00_{\text{basis 2}}$

x_5 = 104 = 1101000 ; $z_5 = 104 \equiv 0 (\text{mod } 2^2) = 00_{\text{basis 2}}$

x_6 = 157 = 10011101 ; $z_6 = 157 \equiv 1 (\text{mod } 2^2) = 01_{\text{basis 2}}$

x_7 = 196 = 11000100 ; $z_7 = 196 \equiv 0 (\text{mod } 2^2) = 00_{\text{basis 2}}$

x_8 = 169 = 10101001 ; $z_8 = 169 \equiv 1 (\text{mod } 2^2) = 01_{\text{basis 2}}$

x_9 = 137 = 10001001 ; $z_9 = 137 \equiv 1 (\text{mod } 2^2) = 01_{\text{basis 2}}$

x_{10} = 168 = 10101000 ; $z_{10} = 168 \equiv 0 (\text{mod } 2^2) = 00_{\text{basis 2}}$

x_{11} = 9 = 1001 ; $z_{11} = 9 \equiv 1 (\text{mod } 2^2) = 01_{\text{basis 2}}$

x_{12} = 81 = 1010001 ; $z_{12} = 81 \equiv 1 (\text{mod } 2^2) = 01_{\text{basis 2}}$

x_{13} = 82 = 1010010 ; $z_{13} = 82 \equiv 2 (\text{mod } 2^2) = 10_{\text{basis 2}}$

x_{14} = 36 = 100100 ; $z_{14} = 36 \equiv 0 (\text{mod } 2^2) = 00_{\text{basis 2}}$

x_{15} = 42 = 101010 ; $z_{15} = 42 \equiv 2 (\text{mod } 2^2) = 10_{\text{basis 2}}$

x_{16} = 92 = 1011100 ; $z_{16} = 92 \equiv 0 (\text{mod } 2^2) = 00_{\text{basis 2}}$

. .

. .

5. Barisan bit acak adalah z_1, z_2, z_3, ..., z_i yang dapat dilihat pada tabel berikut:

Tabel 2. Tabel bit acak z_i

z_i dengan LSB	z_i dengan j
x_1 = 82 ; z_1 = 0	x_1 = 82 = 1010010 $z_1 = 82 \equiv 2 (\text{mod } 2^2) = 10_{\text{basis 2}}$

x_2 = 36; z_2= 0	x_2 = 36 = 100100 z_2= 36 ≡ 0(mod 2^2) = $00_{\text{basis 2}}$
x_3 = 42; z_3= 0	x_3 = 42 = 101010 z_3= 42 ≡ 2(mod 2^2) = $10_{\text{basis 2}}$
x_4 = 92; z_4= 0	x_4 = 92 = 1011100 z_4= 92 ≡ 0(mod 2^2) = $00_{\text{basis 2}}$
x_5 = 104; z_5= 0	x_5 = 104 = 1101000 z_5= 104 ≡ 0(mod 2^2) = $00_{\text{basis 2}}$
x_6 = 157; z_6= 1	x_6 = 157 = 10011101 z_6= 157 ≡ 1(mod 2^2) = $01_{\text{basis 2}}$
x_7 = 196; z_7= 0	x_7 = 196 = 11000100 z_7= 196 ≡ 0(mod 2^2) = $00_{\text{basis 2}}$
x_8 = 169; z_8= 1	x_8 = 169 = 10101001 z_8= 169 ≡ 1(mod 2^2) = $01_{\text{basis 2}}$
x_9 = 137; z_9= 1	x_9 = 137 = 10001001 z_9= 137 ≡ 1(mod 2^2) = $01_{\text{basis 2}}$
x_{10} = 168; z_{10}= 0	x_{10} = 168 = 10101000 z_{10}= 168 ≡ 0(mod 2^2) = $00_{\text{basis 2}}$
x_{11} = 9; z_{11}= 1	x_{11} = 9 = 1001 z_{11}= 9 ≡ 1(mod 2^2) = $01_{\text{basis 2}}$
x_{12} = 81; z_{12}= 1	x_{12} = 81 = 1010001 z_{12}= 81 ≡ 1(mod 2^2) = $01_{\text{basis 2}}$
x_{13} = 82; z_{13}= 0	x_{13} = 82 = 1010010 z_{13}= 82 ≡ 2(mod 2^2) = $10_{\text{basis 2}}$
x_{14} = 36; z_{14}= 0	x_{14} = 36 = 100100 z_{14}= 36 ≡ 0(mod 2^2) = $00_{\text{basis 2}}$
x_{15}= 42; z_{15}= 0	x_{15}= 42 = 101010 z_{15}= 42 ≡ 2(mod 2^2) = $10_{\text{basis 2}}$
x_{16} = 92; z_{16}= 0	x_{16} = 92 = 1011100 z_{16}= 92 ≡ 0(mod 2^2) = $00_{\text{basis 2}}$
.	.
.	.

Sebagai contoh terhadap algoritma Blum Blum Shub yang diterapkan pada algoritma *One Time Pad* akan dijelaskan sebagai berikut:

1. Terdapat Plainteks yang akan dilakukan enkripsi sebagai berikut:
   Hari ini Hujan
   Tiap huruf pada plainteks akan diubah menjadi bilangan desimal sesuai urutannya pada tabel ASCII, sehingga didapat:

   Tabel 3. Tabel Plainteks

Huruf	Decimal
H	72
a	97
r	114
i	105
Spasi	32
i	105
n	110
i	105
Spasi	32
H	72
u	117
j	106
a	97
n	110

2. Pilihlah dua bilangan prima rahasia, *p* dan q, yang masing-masing kongruen dengan 3 mod 4 seperti di persamaan (12):
   i. $p \equiv 3 \bmod 4$ maka $p \bmod 4 = 3$, sehingga diambil $\boldsymbol{p = 11}$
   ii. $q \equiv 3 \bmod 4$ maka $q \bmod 4 = 3$, sehingga diambil $\boldsymbol{q = 19}$
3. Kemudian mencari nilai n dengan mengkalikan *p* dan *q*:
   $n = p.q$
   $n = 11.19$
   $\boldsymbol{n = 209}$

4. Lalu pilih bilangan bulat acak lain, *s*, sebagai umpan sedemikian sehingga:
   i. $2 \leq s < n$
   ii. *s* dan *n* relatif prima

   Relatif prima adalah hasil pemfaktoran dari FPB (Faktor Persekutuan Terbesar) dari 2 bilangan, hanya terdapat satu angka hasil faktor yang sama.

   n = 209, dengan FPB = **1**, 11, 19, 209
   s = 9, dengan FPB = **1**, 3, 9

   Didapat *s* = 9 dan relatif prima dengan *n* = 209, kemudian hitung x_0 :
   $x_0 = s^2 \bmod n$
   $x_0 = 9^2 \bmod 209$
   $x_0 = 81 \bmod 209$

   $x_0 = 81$

5. Barisan bilangan acak dihasilkan dengan banyak sesuai biner-biner yang akan dipakai untuk enkripsi pada plainteks berikut sepanjang yang diinginkan :
   i. Bilangan acak tidak harus 1 bit *LSB* tetapi juga bisa *j* buah bit (*j* adalah bilangan bulat positif yang tidak melebihi ($log_2(log_2$ n)).
      n = p.q
      n = 11.19
      n = 209
      j = 2 *(j tidak melebihi ($log_2(log_2$ 209)) = 2,94623)*
      Maka z_i dari biner, akan maju dua digit LSB dari deret hasil x_i yang diubah menjadi biner.

   ii. Hitung x_i dan cari z_i:

$x_1 = x_0^{\,2} \bmod n = 81^2 \bmod 209 = 82$; z_1 LSB= 0, z_1 j = 10
$x_2 = x_1^{\,2} \bmod n = 82^2 \bmod 209 = 36$; z_2 LSB= 0, z_2 j= 00
$x_3 = x_2^{\,2} \bmod n = 36^2 \bmod 209 = 42$; z_3 LSB= 0, z_3 j = 10
$x_4 = x_3^{\,2} \bmod n = 42^2 \bmod 209 = 92$; z_4 LSB= 0, z_4 j= 00
$x_5 = x_4^{\,2} \bmod n = 92^2 \bmod 209 = 104$; z_5 LSB= 0, z_5 j= 00
$x_6 = x_5^{\,2} \bmod n = 104^2 \bmod 209 = 157$; z_6 LSB= 1, z_6 j= 01
$x_7 = x_6^{\,2} \bmod n = 157^2 \bmod 209 = 196$; z_7 LSB= 0, z_7 j= 00
$x_8 = x_7^{\,2} \bmod n = 196^2 \bmod 209 = 169$; z_8 LSB= 1, z_8 j= 01

$x_9 = x_8^{\,2} \bmod n = 169^2 \bmod 209 = 137$; z_9 LSB= 1, z_9 j= 01
$x_{10} = x_9^{\,2} \bmod n = 137^2 \bmod 209 = 168$; z_{10} LSB= 0, z_{10} j= 00
$x_{11} = x_{10}^{\,2} \bmod n = 168^2 \bmod 209 = 9$; z_{11} LSB= 1, z_{11} j= 01
$x_{12} = x_{11}^{\,2} \bmod n = 9^2 \bmod 209 = 81$; z_{12} LSB= 1, z_{12} j= 01
$x_{13} = x_{12}^{\,2} \bmod n = 81^2 \bmod 209 = 82$; z_{13} LSB= 0, z_{13} j = 10
$x_{14} = x_{13}^{\,2} \bmod n = 82^2 \bmod 209 = 36$; z_{14} LSB= 0, z_{14} j= 00
$x_{15} = x_{14}^{\,2} \bmod n = 36^2 \bmod 209 = 42$; z_{15} LSB= 0, z_{15} j= 10
$x_{16} = x_{15}^{\,2} \bmod n = 42^2 \bmod 209 = 92$; z_{16} LSB= 0, z_{16} j= 00

$x_{17} = x_{16}^{\,2} \bmod n = 92^2 \bmod 209 = 104$; z_5 LSB= 0, z_5 j= 00
$x_{18} = x_{17}^{\,2} \bmod n = 104^2 \bmod 209 = 157$; z_6 LSB= 1, z_6 j= 01
$x_{19} = x_{18}^{\,2} \bmod n = 157^2 \bmod 209 = 196$; z_7 LSB= 0, z_7 j= 00
$x_{20} = x_{19}^{\,2} \bmod n = 196^2 \bmod 209 = 169$; z_{20} LSB= 1, z_{20} j= 01
$x_{21} = x_{20}^{\,2} \bmod n = 169^2 \bmod 209 = 137$; z_{21} LSB= 1, z_{21} j= 01
$x_{22} = x_{21}^{\,2} \bmod n = 137^2 \bmod 209 = 168$; z_{22} LSB= 0, z_{22} j= 00
$x_{23} = x_{22}^{\,2} \bmod n = 168^2 \bmod 209 = 9$; z_{23} LSB= 1, z_{23} j= 01
$x_{24} = x_{23}^{\,2} \bmod n = 9^2 \bmod 209 = 81$; z_{24} LSB= 1, z_{24} j= 01

$x_{25} = x_{24}^{\,2} \bmod n = 81^2 \bmod 209 = 82$; z_{25} LSB= 0, z_{25} j= 10
$x_{26} = x_{25}^{\,2} \bmod n = 82^2 \bmod 209 = 36$; z_{26} LSB= 0, z_{26} j= 00
$x_{27} = x_{26}^{\,2} \bmod n = 36^2 \bmod 209 = 42$; z_{27} LSB= 0, z_{27} j= 10

$x_{28} = x_{27}^{\ 2} \bmod n = 42^2 \bmod 209 = 92$; z_{28} LSB= 0, z_{28} j= 00
$x_{29} = x_{28}^{\ 2} \bmod n = 92^2 \bmod 209 = 104$; z_{29} LSB= 0, z_{29} j= 00
$x_{30} = x_{29}^{\ 2} \bmod n = 104^2 \bmod 209 = 157$; z_{30} LSB= 1, z_{30} j= 01
$x_{31} = x_{30}^{\ 2} \bmod n = 157^2 \bmod 209 = 196$; z_{31} LSB= 0, z_{31} j= 00
$x_{32} = x_{31}^{\ 2} \bmod n = 196^2 \bmod 209 = 169$; z_{32} LSB= 1, z_{32} j= 01

$x_{33} = x_{32}^{\ 2} \bmod n = 169^2 \bmod 209 = 137$; z_{33} LSB= 1, z_{33} j= 01
$x_{34} = x_{33}^{\ 2} \bmod n = 137^2 \bmod 209 = 168$; z_{34} LSB= 0, z_{34} j= 00
$x_{35} = x_{34}^{\ 2} \bmod n = 168^2 \bmod 209 = 9$; z_{35} LSB= 1, z_{35} j= 01
$x_{36} = x_{35}^{\ 2} \bmod n = 9^2 \bmod 209 = 81$; z_{36} LSB= 1, z_{36} j= 01
$x_{37} = x_{36}^{\ 2} \bmod n = 81^2 \bmod 209 = 82$; z_{37} LSB= 0, z_{37} j= 10
$x_{38} = x_{13}^{\ 2} \bmod n = 82^2 \bmod 209 = 36$; z_{38} LSB= 0, z_{38} j= 00
$x_{39} = x_{14}^{\ 2} \bmod n = 36^2 \bmod 209 = 42$; z_{39} LSB= 0, z_{39} j= 10
$x_{40} = x_{15}^{\ 2} \bmod n = 42^2 \bmod 209 = 92$; z_{40} LSB= 0, z_{40} j= 00

$x_{41} = x_{40}^{\ 2} \bmod n = 92^2 \bmod 209 = 104$; z_{41} LSB= 0, z_{41} j= 00
$x_{42} = x_{41}^{\ 2} \bmod n = 104^2 \bmod 209 = 157$; z_{42} LSB= 1, z_{42} j= 01
$x_{43} = x_{42}^{\ 2} \bmod n = 157^2 \bmod 209 = 196$; z_{43} LSB= 0, z_{43} j= 00
$x_{44} = x_{43}^{\ 2} \bmod n = 196^2 \bmod 209 = 169$; z_{44} LSB= 1, z_{44} j= 01
$x_{45} = x_{44}^{\ 2} \bmod n = 169^2 \bmod 209 = 137$; z_{45} LSB= 1, z_{46} j= 01
$x_{46} = x_{45}^{\ 2} \bmod n = 137^2 \bmod 209 = 168$; z_{46} LSB= 0, z_{46} j= 00
$x_{47} = x_{46}^{\ 2} \bmod n = 168^2 \bmod 209 = 9$; z_{47} LSB= 1, z_{47} j= 01
$x_{48} = x_{47}^{\ 2} \bmod n = 9^2 \bmod 209 = 81$; z_{48} LSB= 1, z_{48} j= 01

$x_{49} = x_{48}^{\ 2} \bmod n = 81^2 \bmod 209 = 82$; z_{49} LSB= 0, z_{49} j= 10
$x_{50} = x_{49}^{\ 2} \bmod n = 82^2 \bmod 209 = 36$; z_{50} LSB= 0, z_{50} j= 00
$x_{51} = x_{50}^{\ 2} \bmod n = 36^2 \bmod 209 = 42$; z_{51} LSB= 0, z_{51} j= 10
$x_{52} = x_{51}^{\ 2} \bmod n = 42^2 \bmod 209 = 92$; z_{52} LSB= 0, z_{52} j= 00
$x_{53} = x_{52}^{\ 2} \bmod n = 92^2 \bmod 209 = 104$; z_{53} LSB= 0, z_{53} j= 00
$x_{54} = x_{53}^{\ 2} \bmod n = 104^2 \bmod 209 = 157$; z_{54} LSB= 1, z_{54} j= 01

$x_{55} = x_{54}^{\ 2} \bmod n = 157^2 \bmod 209 = 196$; z_{55} LSB= 0, z_{55} j= 00
$x_{56} = x_{55}^{\ 2} \bmod n = 196^2 \bmod 209 = 169$; z_{56} LSB= 1, z_{56} j= 01

$x_{57} = x_{56}^{\ 2} \bmod n = 169^2 \bmod 209 = 137$; z_{57} LSB= 1, z_{57} j= 01
$x_{58} = x_{57}^{\ 2} \bmod n = 137^2 \bmod 209 = 168$; z_{57} LSB= 0, z_{57} j= 00
$x_{59} = x_{59}^{\ 2} \bmod n = 168^2 \bmod 209 = 9$; z_{59} LSB= 1, z_{59} j= 01
$x_{60} = x_{59}^{\ 2} \bmod n = 9^2 \bmod 209 = 81$; z_{60} LSB= 1, z_{60} j= 01
$x_{61} = x_{60}^{\ 2} \bmod n = 81^2 \bmod 209 = 82$; z_{61} LSB= 0, z_{61} j= 10
$x_{62} = x_{61}^{\ 2} \bmod n = 82^2 \bmod 209 = 36$; z_{62} LSB= 0, z_{62} j= 00
$x_{63} = x_{62}^{\ 2} \bmod n = 36^2 \bmod 209 = 42$; z_{63} LSB= 0, z_{63} j= 10
$x_{64} = x_{63}^{\ 2} \bmod n = 42^2 \bmod 209 = 92$; z_{64} LSB= 0, z_{64} j= 00

$x_{65} = x_{64}^{\ 2} \bmod n = 92^2 \bmod 209 = 104$; z_{65} LSB= 0, z_{65} j= 00
$x_{66} = x_{65}^{\ 2} \bmod n = 104^2 \bmod 209 = 157$; z_{66} LSB= 1, z_{66} j= 01
$x_{67} = x_{66}^{\ 2} \bmod n = 157^2 \bmod 209 = 196$; z_{67} LSB= 0, z_{67} j= 00
$x_{68} = x_{67}^{\ 2} \bmod n = 196^2 \bmod 209 = 169$; z_{68} LSB= 1, z_{68} j= 01
$x_{69} = x_{68}^{\ 2} \bmod n = 169^2 \bmod 209 = 137$; z_{69} LSB= 1, z_{69} j= 01
$x_{70} = x_{69}^{\ 2} \bmod n = 137^2 \bmod 209 = 168$; z_{70} LSB= 0, z_{70} j= 00
$x_{71} = x_{70}^{\ 2} \bmod n = 168^2 \bmod 209 = 9$; z_{71} LSB= 1, z_{71} j= 01
$x_{72} = x_{71}^{\ 2} \bmod n = 9^2 \bmod 209 = 81$; z_{72} LSB= 1, z_{72} j= 01

$x_{73} = x_{72}^{\ 2} \bmod n = 81^2 \bmod 209 = 82$; z_{73} LSB= 0, z_{73} j= 10
$x_{74} = x_{73}^{\ 2} \bmod n = 82^2 \bmod 209 = 36$; z_{74} LSB= 0, z_{74} j= 00
$x_{75} = x_{74}^{\ 2} \bmod n = 36^2 \bmod 209 = 42$; z_{75} LSB= 0, z_{75} j= 10
$x_{76} = x_{75}^{\ 2} \bmod n = 42^2 \bmod 209 = 92$; z_{76} LSB= 0, z_{76} j= 00
$x_{77} = x_{76}^{\ 2} \bmod n = 92^2 \bmod 209 = 104$; z_{77} LSB= 0, z_{77} j= 00
$x_{78} = x_{77}^{\ 2} \bmod n = 104^2 \bmod 209 = 157$; z_{78} LSB= 1, z_{78} j= 01
$x_{79} = x_{78}^{\ 2} \bmod n = 157^2 \bmod 209 = 196$; z_{79} LSB= 0, z_{79} j= 00
$x_{80} = x_{79}^{\ 2} \bmod n = 196^2 \bmod 209 = 169$; z_{80} LSB= 1, z_{80} j= 01

$x_{81} = x_{80}^{\ 2} \bmod n = 169^2 \bmod 209 = 137$; z_{81} LSB= 1, z_{81} j= 01
$x_{82} = x_{81}^{\ 2} \bmod n = 137^2 \bmod 209 = 168$; z_{82} LSB= 0, z_{82} j= 00
$x_{83} = x_{82}^{\ 2} \bmod n = 168^2 \bmod 209 = 9$; z_{83} LSB= 1, z_{83} j= 01
$x_{84} = x_{83}^{\ 2} \bmod n = 9^2 \bmod 209 = 81$; z_{84} LSB= 1, z_{84} j= 01
$x_{85} = x_{84}^{\ 2} \bmod n = 81^2 \bmod 209 = 82$; z_{85} LSB= 0, z_{85} j= 10
$x_{86} = x_{85}^{\ 2} \bmod n = 82^2 \bmod 209 = 36$; z_{86} LSB= 0, z_{86} j= 00
$x_{87} = x_{86}^{\ 2} \bmod n = 36^2 \bmod 209 = 42$; z_{87} LSB= 0, z_{87} j= 10
$x_{88} = x_{87}^{\ 2} \bmod n = 42^2 \bmod 209 = 92$; z_{88} LSB= 0, z_{88} j= 00

$x_{89} = x_{88}^{\ 2} \bmod n = 92^2 \bmod 209 = 104$; z_{89} LSB= 0, z_{89} j= 00
$x_{90} = x_{89}^{\ 2} \bmod n = 104^2 \bmod 209 = 157$; z_{90} LSB= 1, z_{90} j= 01
$x_{91} = x_{90}^{\ 2} \bmod n = 157^2 \bmod 209 = 196$; z_{91} LSB= 0, z_{91} j= 00
$x_{92} = x_{91}^{\ 2} \bmod n = 196^2 \bmod 209 = 169$; z_{92} LSB= 1, z_{92} j= 01
$x_{93} = x_{92}^{\ 2} \bmod n = 169^2 \bmod 209 = 137$; z_{93} LSB= 1, z_{93} j= 01
$x_{94} = x_{93}^{\ 2} \bmod n = 137^2 \bmod 209 = 168$; z_{94} LSB= 0, z_{94} j= 00
$x_{95} = x_{94}^{\ 2} \bmod n = 168^2 \bmod 209 = 9$; z_{95} LSB= 1, z_{95} j= 01
$x_{96} = x_{95}^{\ 2} \bmod n = 9^2 \bmod 209 = 81$; z_{96} LSB= 1, z_{96} j= 01

$x_{97} = x_{96}^{\ 2} \bmod n = 81^2 \bmod 209 = 82$; z_{97} LSB= 0, z_{97} j= 10
$x_{98} = x_{97}^{\ 2} \bmod n = 82^2 \bmod 209 = 36$; z_{98} LSB= 0, z_{98} j= 00
$x_{99} = x_{98}^{\ 2} \bmod n = 36^2 \bmod 209 = 42$; z_{99} LSB= 0, z_{99} j= 10
$x_{100} = x_{99}^{2} \bmod n = 42^2 \bmod 209 = 92$; z_{100} LSB= 0, z_{100} j= 00
$x_{101} = x_{100}^{2} \bmod n = 92^2 \bmod 209 = 104$; z_{101} LSB= 0, z_{101} j= 00
$x_{102} = x_{101}^{2} \bmod n = 104^2 \bmod 209 = 157$; z_{102} LSB= 1, z_{102} j= 01
$x_{103} = x_{102}^{2} \bmod n = 157^2 \bmod 209 = 196$; z_{103} LSB= 0, z_{103} j= 00
$x_{104} = x_{103}^{2} \bmod n = 196^2 \bmod 209 = 169$; z_{104} LSB= 1, z_{104} j= 01

$x_{105} = x_{104}^2 \bmod n = 169^2 \bmod 209 = 137$; z_{105} LSB= 1, z_{105} j= 01
$x_{106} = x_{105}^2 \bmod n = 137^2 \bmod 209 = 168$; z_{106} LSB= 0, z_{106} j= 00
$x_{107} = x_{106}^2 \bmod n = 168^2 \bmod 209 = 9$; z_{107} LSB= 1, z_{107} j= 01
$x_{108} = x_{107}^2 \bmod n = 9^2 \bmod 209 = 81$; z_{108} LSB= 1, z_{108} j= 01
$x_{109} = x_{108}^2 \bmod n = 81^2 \bmod 209 = 82$; z_{109} LSB= 0, z_{109} j= 10
$x_{110} = x_{109}^2 \bmod n = 82^2 \bmod 209 = 36$; z_{110} LSB= 0, z_{110} j= 00
$x_{111} = x_{110}^2 \bmod n = 36^2 \bmod 209 = 42$; z_{111} LSB= 0, z_{111} j= 10
$x_{112} = x_{111}^2 \bmod n = 42^2 \bmod 209 = 92$; z_{112} LSB= 0, z_{112} j= 00

Setelah dilakukan pembangkitan kunci berdasarkan 1 karakter = 8bit dari biner yang dihasilan tiap-tiap z, maka tiap biner siap dipakai sebagai kunci. Untuk tiap biner yang telah dihasilkan akan diubah menjadi desimal untuk proses enkripsi dan dekripsi. Setiap K_i terdiri dari 8 digit biner, sehingga didapat :

Tabel 4. Tabel Kunci K_1 sampai K_3

	LSB		J =2			LSB		J =2			LSB		J =2	
K_1	z_1	0	z_1	1	K_2	z_9	1	z_5	0	K_3	z_{17}	0	z_9	0
	z_2	0		0		z_{10}	0		0		z_{18}	1		1
	z_3	0	z_2	0		z_{11}	1	z_6	0		z_{19}	0	z_{10}	0
	z_4	0		0		z_{12}	1		1		z_{20}	1		0
	z_5	0	z_3	1		z_{13}	0	z_7	0		z_{21}	1	z_{11}	0
	z_6	1		0		z_{14}	0		0		z_{22}	0		1
	z_7	0	z_4	0		z_{15}	0	z_8	0		z_{23}	1	z_{12}	0
	z_8	1		0		z_{16}	0		1		z_{24}	1		1
D	5		136		D	176		17		D	91		69	

Tabel 5. Tabel Kunci K_4 sampai K_6

	LSB		J =2			LSB		J =2			LSB		J =2	
K_4	z_{25}	0	z_{13}	1	K_5	z_{33}	1	z_{17}	0	K_6	z_{41}	0	z_{21}	0
	z_{26}	0		0		z_{34}	0		0		z_{42}	1		1
	z_{27}	0	z_{14}	0		z_{35}	1	z_{18}	0		z_{43}	0	z_{22}	0
	z_{28}	0		0		z_{36}	1		1		z_{44}	1		0
	z_{29}	0	z_{15}	1		z_{37}	0	z_{19}	0		z_{45}	1	z_{23}	0
	z_{30}	1		0		z_{38}	0		0		z_{46}	0		1
	z_{31}	0	z_{16}	0		z_{39}	0	z_{20}	0		z_{47}	1	z_{24}	0
	z_{32}	1		0		z_{40}	0		1		z_{48}	1		1
D	5		136		D	176		17		D	91		69	

Tabel 6. Tabel Kunci K_7 sampai K_9

	LSB		J =2			LSB		J =2			LSB		J =2	
K_7	z_{49}	0	z_{25}	1	K_8	z_{57}	1	z_{29}	0	K_9	z_{65}	0	z_{33}	0
	z_{50}	0		0		z_{58}	0		0		z_{66}	1		1
	z_{51}	0	z_{26}	0		z_{59}	1	z_{30}	0		z_{67}	0	z_{34}	0
	z_{52}	0		0		z_{60}	1		1		z_{68}	1		0
	z_{53}	0	z_{27}	1		z_{61}	0	z_{31}	0		z_{69}	1	z_{35}	0
	z_{54}	1		0		z_{62}	0		0		z_{70}	0		1
	z_{55}	0	z_{28}	0		z_{63}	0	z_{32}	0		z_{71}	1	z_{36}	0
	z_{56}	1		0		z_{64}	0		1		z_{72}	1		1
D	5		136		D	176		17		D	91		69	

Tabel 7. Tabel Kunci K_{10} sampai K_{12}

	LSB		J =2			LSB		J =2			LSB		J =2	
K_{10}	z_{73}	0	z_{37}	1	K_{11}	z_{81}	1	z_{41}	0	K_{12}	z_{89}	0	z_{45}	0
	z_{74}	0		0		z_{82}	0		0		z_{90}	1		1
	z_{75}	0	z_{38}	0		z_{83}	1	z_{42}	0		z_{91}	0	z_{46}	0
	z_{76}	0		0		z_{84}	1		1		z_{92}	1		0
	z_{77}	0	z_{39}	1		z_{85}	0	z_{43}	0		z_{93}	1	z_{47}	0
	z_{78}	1		0		z_{86}	0		0		z_{94}	0		1
	z_{79}	0	z_{40}	0		z_{87}	0	z_{44}	0		z_{95}	1	z_{48}	0
	z_{80}	1		0		z_{88}	0		1		z_{96}	1		1
D	5		136		D	176		17		D	91		69	

Tabel 8. Tabel Kunci K_{13} sampai K_{14}

	LSB		J =2			LSB		J =2	
K_{13}	z_{97}	0	z_{49}	1	K_{14}	z_{105}	1	z_{53}	0
	z_{98}	0		0		z_{106}	0		0
	z_{99}	0	z_{50}	0		z_{107}	1	z_{54}	0
	z_{100}	0		0		z_{108}	1		1
	z_{101}	0	z_{51}	1		z_{109}	0	z_{55}	0
	z_{102}	1		0		z_{110}	0		0
	z_{103}	0	z_{52}	0		z_{111}	0	z_{56}	0
	z_{104}	1		0		z_{112}	0		1
D	5		136		D	176		17	

Berikut adalah proses penyandian atau dapat disebut proses enkripsi, proses dibawah menggunakan algoritma One Time Pad yang dalam bentuk perhitungan ASCII. Pada enkripsi dapat dinyatakan sebagai penjumlahan modulo 256 dari satu karakter plainteks dengan satu karakter kunci one-time pad:

$Ci = (Pi + Ki)$ mod 256

Sedangkan proses dekripsi menggunakan pad yang sama untuk mendekripsikan cipherteks menjadi plainteks dengan persamaan:

$Pi = (Ci - Ki)$ mod 256

Dengan persamaan tersebut, maka akan dillakukan proses enkripsi dari planinteks yang sudah ada di Tabel 3 sebagai berikut :

1. Enkripsi kalimat : Hari ini Hujan

Tabel 9. Tabel Plainteks

Huruf	Decimal ASCII
H	72
a	97
r	114
i	105
Spasi	32
i	105
n	110
i	105
Spasi	32
H	72
u	117
j	106
a	97
n	110

2. Lalu akan dilakukan enkripsi dengan kunci BBS perhitungan LSB, kemudian cipherteks diubah dalam bentuk hexadesimal:

   C_1 = (H + K_1) mod 256

   C_1 = (72 + 5) mod 256 = 77 (desimal) = 4D (hexadesimal)

C_2 = (a + K_2) mod 256

C_2= (97 + 176) mod 256 = 17 (desimal) = 11 (hexadesimal)

C_3 = (r + K_3) mod 256

C_3 = (114+91) mod 256 = 205 (desimal) = CD (hexadesimal)

C_4 = (i + K_4) mod 256

C_4 = (105 + 5) mod 256 = 110 (desimal) = 6E (hexadesimal)

C_5 = (spasi + K_5) mod 256

C_5 = (32+176) mod 256 = 208 (desimal) = D0 (hexadesimal)

C_6 = (i + K_6) mod 256

C_6 = (105+91) mod 256 =196 (desimal) = C4(hexadesimal)

C_7 = (n + K_7) mod 256

C_7 = (110+5) mod 256 =115 (desimal) = 73 (hexadesimal)

C_8 = (i + K_8) mod 256

C_8 = (105+176)mod256= 281 (desimal)= 19(hexadesimal)

C_9 = (spasi + K_9) mod 256

C_9 = (32 + 91) mod 256 = 123 (desimal)= 7B(hexadesimal)

C_{10} = (H + K_{10}) mod 256

C_{10} = (72 + 5) mod 256 = 77 (desimal) = 4D (hexadesimal)

C_{11} = (u + K_{11}) mod 256

C_{11} = (117+ 176)mod 256 = 37 (desimal) = 25(hexadesimal)

C_{12} = (j + K_{12}) mod 256

C_{12} = (106+91) mod 256 = 197 (desimal) = C5 (hexadesimal)

C_{13} = (a + K_{13}) mod 256

C_{13} = (97+ 5) mod 256 = 102 (desimal) = 66 (hexadesimal)

C_{14} = (n + K_{14}) mod 256

C_{14} = (110+176)mod 256 = 30 (desimal) = 1E(hexadesimal)

3. Lalu akan dilakukan enkripsi dengan kunci BBS perhitungan *j* buah bit, kemudian cipherteks dalam bentuk hexadesimal:

   C_1 = (H + K_1) mod 256

   C_1 = (72+136)mod 256 = 208 (desimal) = D0 (hexadesimal)

   C_2 = (a + K_2) mod 256

   C_2 = (97 + 17) mod 256 = 114 (desimal) = 72 (hexadesimal)

   C_3 = (r + K_3) mod 256

   C_3 = (114+69) mod 256 = 183 (desimal) = B7 (hexadesimal)

   C_4 = (i + K_4) mod 256

   C_4 = (105 + 5) mod 256 = 110 (desimal) = F1 (hexadesimal)

   C_5 = (spasi + K_5) mod 256

   C_5 = (32+17) mod 256 = 49 (desimal) = 31 (hexadesimal)

   C_6 = (i + K_6) mod 256

   C_6 = (105+69)mod 256 = 174 (desimal) = AE (hexadesimal)

   C_7 = (n + K_7) mod 256

   C_7 = (110+136)mod 256= 115 (desimal) = F6 (hexadesimal)

C_8 = (i + K_8) mod 256

C_8 = (105+17)mod256= 122 (desimal) = 7A (hexadesimal)

C_9 = (spasi + K_9) mod 256

C_9 = (32 + 69) mod 256 = 101 (desimal) = 65 (hexadesimal)

C_{10} = (H + K_{10}) mod 256

C_{10} = (72 + 36) mod 256 = 208 (desimal) = D0 (hexadesimal)

C_{11} = (u + K_{11}) mod 256

C_{11} = (117+ 17)mod 256 = 134 (desimal) = 86(hexadesimal)

C_{12} = (j + K_{12}) mod 256

C_{12} = (106+69)mod 256 = 175 (desimal) = AF (hexadesimal)

C_{13} = (a + K_{13}) mod 256

C_{13} = (97+136)mod 256 = 233 (desimal) = E9 (hexadesimal)

C_{14} = (n + K_{14}) mod 256

C_{14} = (110+17)mod 256 = 127 (desimal) = 7F (hexadesimal)

4. Didapat cipherteks dari enkripsi masing-masing:
   i. Cipherteks hasil enkripsi *One Time Pad* dengan pembangkit kunci Blum Blum Shub perhitungan LSB dalam hexadesimal : 4D11CD6ED0C473197B4D25C5661E

   ii. Cipherteks hasil enkripsi *One Time Pad* dengan pembangkit kunci Blum Blum Shub perhitungan *j* buah bit dalam hexadesimal : D072B7F131AEF67A65D086AFE97F

5. Untuk melakukan dekripsi terhadap cipherteks yang sudah terbentuk dapat dilakukan cara sebagai berikut:

   A. Dekripsi dengan kunci BBS perhitungan LSB, kemudian cipherteks dalam hexadesimal diubah dalam bentuk desimal untuk menjadi karakter:

C_1 = 4D (hexadesimal) = 77 (desimal)

P_1 = $(C_1 - K_1)$ mod 256 = (77 - 5) mod 256 = 72 = H

C_2 = 11 (hexadesimal) = 17 (desimal),

sedangkan K_2 = 176, maka $K_2 > C_2$

P_2 = $(C_2 - K_2)$ mod 256 = (17 - 176) mod 256

P_2 = -159 mod 256 → -159 = 256.(-1) + P_2

P_2 = 97

P_2 = $(C_2 - K_2)$ mod 256 = a

C_3 = CD (hexadesimal) = 205 (desimal)

P_3 = $(C_3 - K_3)$ mod 256 = (205-91) mod 256 = 114 = r

C_4 = 6E (hexadesimal) = 110 (desimal)

P_4 = $(C_4 - K_4)$ mod 256 = (110 - 5) mod 256 = 105 = i

C_5 = D0 (hexadesimal) = 208 (desimal)

P_5 = $(C_5 - K_5)$ mod 256 = (208-176)mod256 =32= spasi

C_6 = C4 (hexadesimal) =196 (desimal)

P_6 = $(C_6 - K_6)$ mod 256 = (196-91) mod 256 = 105 = i

C_7 = 73 (hexadesimal) = 115(desimal)

P_7 = $(C_7 - K_7)$ mod 256 = (115-5) mod 256 = 110 = n

C_8 = 19 (hexadesimal) = 25 (desimal),
sedangkan K_8 = 176, maka $K_8 > C_8$,
P_8 = $(C_8 - K_8)$ mod 256 = (25 - 176) mod 256
P_8 = -151 mod 256 → -151 = 256.(-1) + P_8
P_8 = 105
P_8 = $(C_8 - K_8)$ mod 256 = i

C_9 = 7B (hexadesimal) = 123 (desimal)
P_9 = $(C_9 - K_9)$ mod 256 = (123-91)mod 256=32= spasi

C_{10} = 4D (hexadesimal) = 77 (desimal)
P_{10} = $(C_{10} - K_{10})$ mod 256 = (77 - 5) mod 256 = 72 = H

C_{11} = 25 (hexadesimal) = 37 (desimal),
sedangkan K_{11} = 176, maka $K_{11} > C_{11}$
P_{11} = $(C_{11} - K_{11})$ mod 256 = (37 - 176) mod 256
P_{11} = -139 mod 256 → -139 = 256.(-1) + P_{11}
P_{11} = 117
P_{11} = $(C_{11} - K_{11})$ mod 256 = u

C_{12} = C5 (hexadesimal) =197 (desimal)
P_{12} = $(C_{12} - K_{12})$ mod 256 = (197-91) mod 256 = 106 = j

C_{13} = 66 (hexadesimal) = 102 (desimal)
P_{13} = $(C_{13} - K_{13})$ mod 256 = (102- 5) mod 256 = 97 = a

C_{14} = 1E (hexadesimal) =30 (desimal),
sedangkan K_{14} = 176, maka $K_{14} > C_{14}$
P_{14} = $(C_{14} - K_{14})$ mod 256 = (30 - 176) mod 256

P_{14} = -146 mod 256 → -146 = 256.(-1) + P_{14}

P_{14} = 110

P_{14} = (C_{14} - K_{14}) mod 256 = n

B. Dekripsi dengan kunci BBS perhitungan *j* buah bit, kemudian cipherteks dalam hexadesimal diubah dalam bentuk desimal untuk menjadi karakter:

C_1 = D0 (hexadesimal) = 208 (desimal)

P_1 = (C_1- K_1) mod 256 = (208 - 136) mod 256= 72 = H

C_2 = 72 (hexadesimal) = 114 (desimal),

P_2 = (C_2- K_2) mod 256 = (114 - 17) mod 256 = 97 = a

C_3 = B7 (hexadesimal) = 183 (desimal)

P_3 = (C_3 - K_3) mod 256 = (183-69) mod 256 = 114 = r

C_4 = F1 (hexadesimal) = 241 (desimal)

P_4 = (C_4 - K_4) mod 256 = (241 - 136) mod 256 = 105 = i

C_5 = 31 (hexadesimal) = 49 (desimal)

P_5 = (C_5 - K_5) mod 256 = (49-17)mod 256 = 32 = spasi

C_6 = AE (hexadesimal) =174 (desimal)

P_6 = (C_6 - K_6) mod 256 = (174-69) mod 256 = 105 = i

C_7 = F6 (hexadesimal) = 246 (desimal)

P_7 = (C_7 - K_7) mod 256 = (246-136) mod 256 = 110 = n

C_8 = 7A (hexadesimal) = 122 (desimal),

P_8 = (C_8 - K_8) mod 256 = (122-17)mod 256 = 105 = i

C_9 = 65 (hexadesimal) = 101 (desimal)

P_9 = $(C_9 - K_9)$ mod 256 = (101-69)mod 256=32= spasi

C_{10} = D0 (hexadesimal) = 208 (desimal)

P_{10} = $(C_{10} - K_{10})$ mod 256 = (208 - 136) mod 256 = 72 = H

C_{11} = 86 (hexadesimal) = 134 (desimal),

P_{11} = $(C_{11}-K_{11})$ mod 256 = (134- 17)mod 256 = 117 = u

C_{12} = AF (hexadesimal) =175 (desimal)

P_{12} = $(C_{12} - K_{12})$ mod 256 = (175-69) mod 256 = 106 = j

C_{13} = E9 (hexadesimal) = 233 (desimal)

P_{13} = $(C_{13} - K_{13})$ mod 256 = (233- 136) mod 256 = 97 = a

C_{14} = 7F (hexadesimal) =127 (desimal),

P_{14} = $(C_{14}-K_{14})$ mod 256 = (127-17)mod 256 = 110 = n

6. Didapat plainteks dari hasil dekripsi masing-masing:
   i. Cipherteks hasil enkripsi *One Time Pad* dengan pembangkit kunci Blum Blum Shub perhitungan LSB dalam hexadesimal :
   4D11CD6ED0C473197B4D25C5661E

   Dari proses dekripsi didapat plainteks :
   Hari ini Hujan

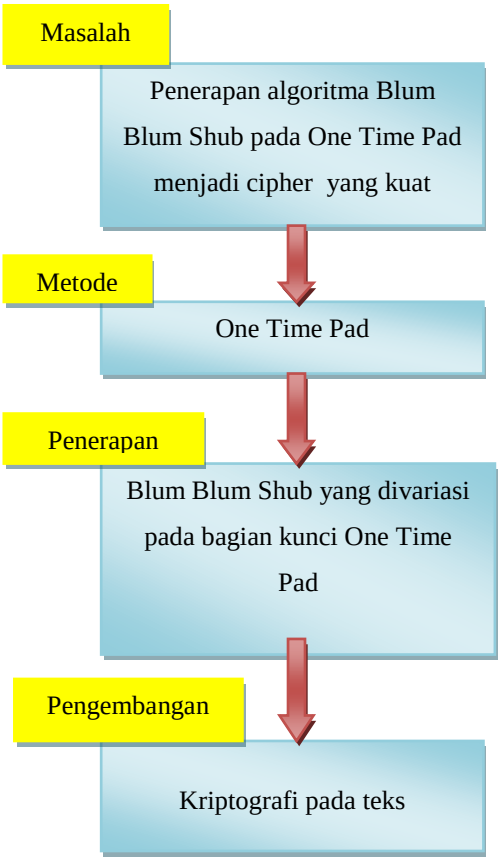
   ii. Cipherteks hasil enkripsi *One Time Pad* dengan pembangkit kunci Blum Blum Shub perhitungan *j* buah bit dalam hexadesimal :
   D072B7F131AEF67A65D086AFE97F

Dari proses dekripsi didapat plainteks :

Hari ini Hujan

2.2 Kerangka Pemikiran

Penelitian ini bertujuan untuk meningkatkan kekuatan kedua teknik algoritma yaitu *One Time Pad* dengan pembangkit kunci berdasarkan algoritma *Blum Blum Shub (BBS)*. Dengan teknik *Blum Blum Shub* (*BBS*), kunci dapat dibangkitkan secara acak yang merupakan salah satu syarat untuk membuat algoritma yang aman di dalam kriptografi. Setelah kedua teknik algoritma tersebut masing-masing berdasarkan algoritma *Blum Blum Shub* (*BBS*) pada pembangkitan kunci yanga digambarkan pada gambar diberikut.



Gambar 6 : Kerangka pemikiran