BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Terdahulu

Peneliti yang pertama yang dilakukan oleh Salma Dola Silalahi, Guidio Leonarda Ginting (2020), Universitas Budi Darma, Medan Pada Jurnal Penelitian yang berjudul "Penerapan Algoritma Kriptografi Word Auto Key Encryption Untuk Pengamanan Soal Ujian Berbasis Komputer" Ujian Akhir Semester (UAS) berbasis komputer yang dilaksanakan di SMK Swasta Dwiwarna Medan untuk saat ini hanya soal ujian jurusan Teknik Komputer dan Jaringan dan Rekayasa Perangkat Lunak dengan mode ujian pilihan berganda. Aplikasi yang digunakan untuk menampilkan soal ujian berbasis web dan soal ujian disimpan ke dalam database. Ujian online berbasis web dimulai dengan peserta masuk/login ke aplikasi ujian online kemudian mengerjakan semua soal yang diujiankan dan pengumpulan jawaban ujian dilakukan dengan peserta mensubmit semua jawaban yang telah dipilih ke dalam server. Kemudian server akan memeriksa hasil ujian siswa dan mendapatkan score berupa hasil jawaban yang benar atau jumlah jawaban yang salah. Masalah yang pernah terjadi dengan ujian berbasis web adalah ketika proses penyimpanan soal ke dalam database yang dilakukan oleh admin. Soal yang sudah disimpan dapat dibaca dengan mudah oleh siswa di interface web admin, hal tersebut terjadi karena sistem web belum mempunyai keamanan enkripsi database soal yang mengakibatkan soal dapat dibaca dengan jelas. Masalah lain yang terjadi adalah sistem ujian berbasis

web pada SMK Swasta Dwiwarna hanya mengandalkan server tanpa pengamanan lain, dimana pada dasarnya komputer server tidak memeliki konfigurasi keamanan database. Database soal ujian yang terletak di dalam komputer dapat dengan mudah diakses oleh pihak yang tidak bertanggung jawab jika hanya mengandalkan keamanan dasar dari komputer itu sendiri. Sehingga dibutuhkan sebuah teknik keamanan kriptografi yang dapat meminalisir penyadapan database yang dapat dibaca oleh manusia.

Penelitian kedua yang dilakukan oleh Peter Jaya, Safitri Juanita (2018), Dari Fakultas Teknologi informasi Universitas Budi luhur pada jurnal penelitian yang berjudul "Aplikasi Pengamanan Basis Data Dengn Algoritma RSA dan WAKE Berbasis Desktop" Menjelaskan bahwa banyak pelanggan yang datanya tersimpan dalam bentuk file basis data bernama data pelanggan. Data pelanggan merupakan data data yang penting yang berkaitan dengan identitas pelanggan dan bersifat rahasia memungkinkan untuk diambil atau dirusak oleh pihak yang menginginkan data tersebut, sehingga dibutuhkan perangkat lunak untuk mengamankan data tersebut. Metode yang digunakan untuk membangun perangkat lunak adalah metode waterfall. Bahasa pemrograman yang digunakan adalah java, dengan basis data MySQL. Aplikasi ini mengimplementasikan metode kriptografi dengan algoritma RSA dan WAKE, Algoritma RSA yang digunakan untuk proses Enkripsi dan Dekripsi data dan algoritma WAKE yang digunakan untuk proses pembuatan public key dan private key pada saat melakukan generate key. Aplikasi ini menggunakan algoritma RSA dan WAKE yang digunakan untuk proses enkripsi dan dekripsi basis data. Diimplementasikan menggunkan bahasa pemprograman

java dengan menggunakan Algoritma RSA dan WAKE untuk proses enkripsi dan Dekripsi.

II.2. Landasan Teori

Berikut ini adalah beberapa landasan teori yang digunakan dalam pembuatan skripsi ini.

II.2.1 Aplikasi

Aplikasi merupakan program yang dirancang untuk melakukan fungsi yang digunakan oleh pengguna ataupun pada aplikasi lainnya. Aplikasi juga merupakan suatu program yang dibuat pengguna untuk mengerjakan tugas tertentu. Aplikasi dijelaskan juga sebagai penerapan maupun penggunaan sebuah rancangan pengkajian utama atau program komputer yang dibuat untuk membantu tugas manusia. Lebih lanjut dijelaskan bahwa aplikasi perangkat lunak yang dirancang untuk penggunaan pelaku tertentu. (Stevanus, 2020 : 29)

II.2.2 *Desktop*

Desktop merupakan sebuah aplikasi yang mampu beroperasi sendiri tanpa memakai suatu koneksi internet dengan sistem operasi atau program khusus. Aplikasi *desktop* diutamakan pada aplikasi yang berdiri sendiri. Tujuannya yaitu mempermudahkan pengguna mengatur kembali aplikasi untuk memaksimalkan tenaga, waktu sekaligus pengeluaran. Aplikasi berbasis *desktop* dapat dibedakan 2 jenis. Yaitu pemrograman konvensional dan pemrograman visual. Pemrograman konvensional yaitu suatu metode merancang aplikasi agar dapat menerapkan baris

perbaris kode namun memerlukan waktu lama. Pemrograman visual yaitu suatu metode membuat program dengan melakukan koneksi antar objek-objek. Cara melakukannya yaitu dengan mengambar atau menunjuk diagram dan gambaran yang berhubungan dengan diagram jalur. (Stevanus, 2020 : 30)

II.2.3. Database

Database merupakan kumpulan *File-file* yang saling berkaitan dan berinteraksi, relasi tersebut bila ditunjukan dengan kunci dari tiap-tiap *file* yang ada. Satu *database* menunjukkan suatu kumpulan data yang dipakai dalam suatu lingkup perusahaan, instansi,. Pengolahan *database* merupakan suatu cara yang dilakukan terhadap *file-file* yang berada di suatu instansi yang mana *file* tersebut dapat disusun, diurut, diambil sewaktu-waktu serta dapat ditampilkan dalam bentuk suatu laporan sehingga dapat mengolah *file-file* yang berisikan informasi tersebut secara rapi. (Muhammad Yusuf, 2019 : 18)

II.2.4. Kriptografi

Kriptografi merupakan studi terhadap penyandian suatu tulisan, database dan informasi rahasia dengan menggunakan suatu teknik matematis. Dengan menggunakan kriptografi, tulisan, databse dan informasi dapat disandikan kedalam bentuk yang tidak bisa dimengerti sehingga informasi tersebut tidak dapat di akses oleh orang orang tidak berkepentingan.

Adapun teknik kriptografi adalah sebagai berikut :

1. Cipher Klasik

   Cipher klasik adalah tipe *cipher* yang paling dasar, yang beroperasi pada huruf (A-Z), tipikalnya adalah penukaran atau penggeseran huruf. Implementasi cipher ini secara umum bisa dilaksanakan baik dengan tangan atau dengan perangkat mekanis sederhana. Karena sandi ini mudah diuraikan maka cipher klasik ini umumnya tidak dapat diandalkan. Contoh tipe cipher klasik:

   a. *Caesar Cipher*
   b. *Vignere Cipher*
   c. ROT13

2. Cipher Modern

   Desain *cipher modern* membantu menahan berbagai serangan. *Cipher modern* memberikan kerahasiaan pesan, integritas, dan otentikasi pengirim. Pengguna dapat mengandalkan *cipher modern* dengan bantuan fungsi matematika satu arah yang mampu memfaktorkan bilangan prima yang besar.

   Contoh tipe *cipher modern* berdasarkan key yang digunakan:

   a. *Algoritma symmetric key* (*Private-key cryptography*): Menggunakan key yang sama untuk melakukan enkripsi dan dekripsi
   b. *Algoritma assymetric key* (*Public-key cryptography*): Menggunakan key yang berbeda antara enkripsi dan dekripsi

c. *Block Cipher* Algoritma yang beroperasi pada blok (kelompok bit) ukuran tetap dengan transformasi yang tidak bervariasi dan ditentukan oleh kunci simetrik. Kebanyakan *cipher modern* adalah *cipher block*. Ini banyak digunakan untuk mengenkripsi data massal. Contohnya termasuk DES, AES, IDEA, dll.

d. *Stream Cipher*

Symmetric-key yang merupakan *digit plaintext* yang dikombinasikan dengan aliran kunci (aliran digit cipher pseudorandom). Di sini, pengguna menerapkan kunci ke setiap bit, satu per satu. Contohnya termasuk RC4, SEAL, dll.

II.2.4.1. Kriptografi Klasik

Kriptografi klasik adalah kriptografi dalam pembuatannya maupun analisisnya sama sekali tidak melibatkan komputer atau perangkat mesin. Alat-alat yang digunakan berkutat pada pemanfaatan kertas, pena, batu, serta alat-alat lain yang tidak tergolong dalam perangkat mesin modern sama sekali. Ciri khas dari kriptografi klasik ialah lebih berbasis pada karakter, baik karakter tulisan maupun karakter pesan yang disampaikan. Ciri lainnya berupa penggunaan alat-alat yang masih terbilang tradisional karena pada waktu kemunculannya belum mengenal komputer yaitu menggunakan kertas dan pena. Termasuk ke dalam kriptografi kunci simetris. Semua alogaritma kriptografi (chipper) dari kriptografi klasik termasuk dalam sistem kriptografi yang bersistem simetris. Teknik enkripsi pada kriptografi klasik semuanya sama seperti kunci enkripsi. Maksudnya, untuk

memahami sebuah teks tersembunyi dapat dilakukan secara serupa seperti saat pembuatannya. Contoh kriptografi klasik adalah sebagai berikut :

a. Metode Vigenere

   Vigenere Cipher, merupakan pengembangan dari Caesar Cipher merupakan metode menyandikan teks alfphabet dengan menggunakan sederetan sandi caesar berdasarkan huruf huruf pada kata kunci. Jadi penyandian dengan metode ini juga sudah menggunakan kunci.Enkripsi (penyandian) dengan sandi Vigenere dapat ditulis secara matematis, dengan menggunakan penjumlahan dan operasi modulus dan dekripsi, yaitu : Keteranga n : Ci adalah huruf ke-i pada teks tersandi, Pi adalah huruf ke-i pada teks terang, Ki adalah huruf ke-i pada kata kunci, dan mod adalah operasi modulus (sisa pembagian).

II.2.4.2. Kriptografi Modern

Kriptografi modern merupakan suatu perbaikan yang mengacu pada kriptografi klasik. Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer. Algoritma kriptografi modern umumnya beroperasi dalam mode bit. Berbeda dengan kriptografi klasik yang beroperasi dalam mode karakter (seperti yang dilakukan pada cipher substitusi atau cipher transposisi dari algoritma kriptografi klasik). Operasi dalam mode bit berarti semua data dan informasi (baik kunci, plainteks, maupun cipherteks) dinyatakan dalam rangkaian (string) bit biner, 0 dan 1. Algoritma enkripsi dan dekripsi memproses semua data dan informasi

dalam bentuk rangkaian bit. Rangkaian bit yang menyatakan plainteks dienkripsi menjadi cipherteks dalam bentuk rangkaian bit, demikian sebaliknya.

a. Macam-Macam Kriptografi Modern

Kriptografi modern merupakan suatu perbaikan yang mengacu pada kriptografi klasik. Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer. Algoritma kriptografi modern terdiri dari tiga bagian :

1. Algoritma Simetris

Algoritma simetris adalah algoritma yang menggunakan kunci yang sama untuk enkripsi dan deskripsinya. Algoritma kriptografi simetris sering disebut algoritma kunci rahasia, algoritma kunci tunggal atau algoritma satu kunci dan megharuskan pengirim dan penerima menyetujui suatu kunci tersebut. Kelebihan dari kriptografi simetris waktu proses untuk enkripsi dan deskripsi relatif cepat. Hal ini disebabkan efesiensi yang terjadi pada pembangkit kunci. Maka proses relative cepat maka algoritma ini tepat untuk digunakan pada sistem komunikasi digital secara *real time* seperti GSM.

Contoh Alice ingin mengirim pesan x dengan aman menggunakan saluran umum kepada Bob. Alice menggunakan kunci xO yang sebelumnya telah disepakati Alice dan Bob. Untuk mengirim pesan e XO (x) kepada Bob, dia akan deskripsi teks kode yang diterima dengan

kunci yang sama dengan yang digunakan untuk memperoleh akses ke pesan yang diterima. Begitu juga sebaliknya. Aplikasi dari algoritma simetris digunakan oleh beberapa algoritma dibawah ini :

a. Data Encryption Standar (DES)
b. Advance Encryption Standar (AES)
c. International Data Encryption Algoritma
d. A5
e. RC4

2. Algoritma Asimetris

   Algoritma asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan satu lagi deskripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsi suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia itu, dalam hal ini kunci rahasia, untuk melakukan pembongkaran terhadap kode yang dikirim untuknya. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA (merupakan singkatan dari nama penemunya, yaitu Rivest, Shamir dan Adleman).

3. Algoritma Hibrida

   Algoritma hibrida adalah algoritma yang memanfaatkan dua tingkatan kunci, yaitu kunci rahasia (simetri) yang disebut juga session key (kunci sesi) untuk enkripsi data dan pasangan kunci rahasia adalah kunci publik untuk pemberian tanda tangan digital serta melindungi kunci simetris.

Algoritma Kriptografi yang beroperasi dalam mode bit dapat dikelompokan menjadi dua kategori, yaitu :

1. Cipher Aliran Algoritma kriptografi beroperasi pada *plainteks/cipherteks* dalam bentuk bit tunggal, yag dalam hal ini rangkaian rangkaian bit dienkripsikan / dideskripsikan bit per bit. *Stream Chiper* atau *Stream Encryption* merupakan suatu teknik enkripsi data dengan cara melakukan transformasi dari tiap bit secara terpisah berdasarkan posisi tiap bit dalam aliran data yang biasanya dikendalikan menggunakan operasi XOR. Enkripsi aliran data merupakan hasil dari operasi XOR setiap bit *plainteks* dengan setiap bit kuncinya. Pada stream chiper bila terjadi kesalahan selama transisi maka kesalahan pada teks enkripsi penerima akan terjadi tepat ditempat kesalahan tersebut terjadi. Dalam praktek pertimbangan kesalahan yang mungkin terjadi sangatlah penting untuk penentuan teknik enkripsi yang akan digunakan.
2. Cipher Blok (Block Cipher) Algoritma kriptografi beroperasi pada *plainteks/cipherteks* dalam bentuk blok bit, yang dalam hal ini rangkaian bit dibagi menjadi blok-blok bit yang panjangnya sudah ditentukan sebelumnya.

Misalnya panjang blok adalah 64 bit, maka itu algoritma enkripsi memperlakukan 8 karakter setiap kali penyandian (1 karakter = 8 bit dalam pengkodean ASCII)

b. Rangkaian bit pada kriptografi modern

Rangkaian bit yang dipecah menjadi blok-blok bit dapat ditulis dalam sejumlah cara bergantung pada panjang blok.

Contoh: Plainteks 100111010110 dibagi menjadi blok bit yang panjangnya 4 menjadi

1001 1101 0110

Setiap blok menyatakan bilangan bulat dari 0 sampai 15, yaitu 9 13 6 Bila *plainteks* dibagi menjadi blok-blok yang berukuran 3 bit, maka rangkaian bit di atas menjadi:

100 111 010 110

Setiap blok menyatakan bilangan bulat dari 0 sampai 7, yaitu 4 7 2 6 Bila panjang rangkaian bit tidak habis dibagi dengan ukuran blok yang ditetapkan, maka blok yang terakhir ditambah dengan bit-bit semu yang disebut *padding bits*.

Misalnya rangkaian bit di atas dibagi menjadi blok 5-bit menjadi

10011 10101 **000**10

Blok yang terakhir telah ditambahkan 3 bit 0 di bagian awal (dicetak tebal) agar ukurannya menjadi 5 bit. *Padding bits* dapat mengakibatkan ukuran plainteks hasil dekripsi lebih besar daripada ukuran plainteks semula.

Cara lain untuk menyatakan rangkaian bit adalah dengan notasi heksadesimal (HEX). Rangkaian bit dibagi menjadi blok yang berukuran 4

bit dengan representasi dalam HEX adalah: 0000 = 0 0001 = 1 0010 = 2 0011 = 3

0100 = 4 0101 = 5 0011 = 6 0111 = 7

1000 = 8 1011 = 9 1010 = A 1011 = B

1100 = C 1101 = D 1101 = E 1111 = F

Misalnya, plainteks 100111010110 dibagi menjadi blok bit yang panjangnya 4 menjadi

1001 1101 0110

yang dalam notasi HEX adalah 9 D 6

Operator XOR

- Operator biner yang sering digunakan dalam *cipher* yang yang beroperasi dalam mode bit adalah XOR atau *exclusive-or*.
- Notasi matematis untuk opeartor XOR adalah $\oplus$ (dalam Bahas C, operator XOR dilambangkan dengan ^).
- Operator XOR diperasikan pada dua bit dengan aturan sebagai berikut:

  $0 \oplus 0 = 0$

  $0 \oplus 1 = 1$

  $1 \oplus 0 = 1$

  $1 \oplus 1 = 0$

- Operator XOR identik dengan penjumlahan modulo 2.

  Misalkan *a*, *b*, dan *c* adalah peubah Boolean. Hukum-hukum yang terkait dengan operator XOR:

  (i) $a \oplus a = 0$

(ii) $a \oplus b = b \oplus a$ (Hukum komutatif)

(iii) $a \oplus (b \oplus c) = (a \oplus b) \oplus c$ (Hukum asosiatif)

- Jika dua rangkaian dioperasikan dengan XOR, maka operasinya dilakukan dengan meng-XOR-kan setiap bit yang berkoresponden dari kedua ramngkaian bit tersebut.

  Contoh: 10011 ⊕ 11001 = 01010 yang dalam hal ini, hasilnya diperoleh sebagai berikut:

  1 0 0 1 1

  1 1 0 0 1 ⊕

  1 ⊕ 1 0 ⊕ 1 0 ⊕ 0 1⊕ 0 1 ⊕ 1

  0 1 0 1 0

- Algoritma enkripsi sederhana yang menggunakan XOR adalah dengan meng-XOR-kan plainteks (*P*) dengan kunci (*K*) menghasilkan cipherteks:

  $C = P \oplus K$ (6.1)

  Karena meng-XOR-kan nilai yang sama dua kali menghasilkan nilai semula, maka proses dekripsi menggunakan persamaan:

  $P = C \oplus K$ (6.2)

  Contoh: plainteks 01100101 (karakter ‘e’)

  kunci 00110101 ⊕ (karakter ‘5’)

  cipherteks 01010000 (karakter ‘P’)

  kunci 00110101 ⊕ (karakter ‘5’)

  plainteks 01100101 (karakter ‘e’)

- Program komersil yang berbasis DOS atau Macintosh menggunakan algoritma XOR sederhana ini. Sayangnya, algoritma XOR sederhana tidak aman karena*cipherteks*nya mudah dipecahkan.

Cara memecahkannya adalah sebagai berikut (asumsi: panjang kunci adalah sejumlah kecil *byte*):

- Cari panjang kunci dengan prosedur *counting coincidence* sbb: XOR-kan cipherteks terhadap dirinya sendiri setelah digeser sejumlah *byte*, dan hitung jumlah *byte* yang sama. Jika pergeseran itu kelipatan dari panjang kunci (yang tidak diketahui), maka 6% dari *byte* akan sama. Jika tidak, maka 0.4% akan sama. Angka persentase ini disebut *index of coincidence*. Pergeseran terkecil mengindikasikan panjang kunci yang dicari.
- Geser cipherteks sejauh panjang kunci dan XOR-kan dengan dirinya sendiri. Operasi ini menghasilkan plainteks yang ter-XOR dengan plainteks yang digeser sejauh panjang kunci tersebut.Cari panjang kunci dengan prosedur*counting coincidence* sbb: XOR-kan cipherteks terhadap dirinya sendiri setelah digeser sejumlah *byte*, dan hitung jumlah *byte* yang sama. Jika pergeseran itu kelipatan dari panjang kunci (yang tidak diketahui), maka 6% dari *byte* akan sama. Jika tidak, maka 0.4% akan sama. Angka persentase ini disebut *index of coincidence*. Pergeseran terkecil mengindikasikan panjang kunci yang dicari.

- Geser cipherteks sejauh panjang kunci dan XOR-kan dengan dirinya sendiri. Operasi ini menghasilkan plainteks yang ter-XOR dengan plainteks yang digeser sejauh panjang kunci tersebut.

II.2.4.3. Kunci Publik dan Kunci Private Pada Kriptogrfi

Public key dan private key merupakan pasangan kunci yang bertugas dalam proses enkripsi dan dekripsi. Jadi saat private key melakukan enkripsi data maka public key digunakan untuk mendekripsi data, hal tersebut salah satu alasan mengapa private key tidak dapat diberikan ke sembarang orang. Public key dan Private key melakukan proses enkripsi dan dekripsi dengan cara menyandikan informasi. Kedua key (kunci) tersebut bekerja pada dua sistem enkripsi yaitu simetris dan asimetris. Enkripsi simetris biasa disebut juga dengan private key atau private key encryption. Dalam sistem simetris proses enkripsi dan dekripsi dilakukan dengan key yag sama. Sedangkan asimetris menggunakan sepasang key seperti public key dan private key untuk keamanan yang lebih baik. Proses ini melibatkan kedua key, saat public key melakukan enkripsi maka private key melakukan dekripsi. Hal tersebut dilakukan untuk memastikan bahwa data terlindungi selama transmisi berlangsung.

a. Public Key

   Public key menggunakan algoritma asimetris yang mengubah pesan menjadi format yang tidak dapat dibaca. Seseorang yang memiliki public key dapat melakukan enkripsi pesan yang ditujukan pada penerima tertentu. Penerima dengan private key hanya dapat mendekode pesan yang telah

dienkripsi public key. Key yang tersedia melalui direktori yang dapat diakses oleh public.

b. Private key

Private key adalah kunci rahasia yang digunakan untuk mendekripsi pesan. Dalam metode tradisional, private key hanya dibagikan dalam komunikator untuk mengaktifkan enkripsi dan dekripsi pesan, namun jika kunci tersebut hilang system akan menjadi batal. Untuk menghindari hal tersebut PKI (Public Key Infrastructure) memberikan kebijakan dimana public key dapat digunakan bersama dengan private key. PKI memungkinkan pengguna internet untuk bertukar informasi dengan cara aman dengan menggunakan public key dan private key.

II.2.5. Algoritma Word Auto Key Encryption (WAKE)

Algoritma WAKE merupakan Salah satu metode yang telah digunakan secara komersial. Algoritma WAKE merupakan singkatan dari Word Auto Key Encryption ditemukan pertama kali oleh David J. Wheeler pada tahun 1993. Dan merupakan salah satu algoritma kriptogtafi modern yang beroperasi dalam mode bit. Operasi dalam mode bit berarti semua data dan informasi (baik kunci, plaintext maupun ciphertext) dinyatakan dalam rangakian (String) bit binner, 0 dan 1. Algoritma enkripsi dan dekripsi memproses semua data dan informasi dalam bentuk bit rangkaian bit yang menyatakan plaintext dienkripsi menjadi ciphertext dalam bentuk rangkaian bit,

Algoritma stream cipher yang cepat dalam implementasinya dalam perangkat lunak. Metode ini menggunakan kunci 128 bit, plaintext 32 bit dan sebuah tabel 256 x 32 bit. Dalam algoritmanya metode ini menggunakan operasi XOR, AND, OR dan Shift Right. Metode ini dapat dibagi menjadi beberapa proses yaitu proses pembentukan tabel dan kunci, enkripsi dan dekripsi. Proses penyelesaian metode ini cukup rumit dan sulit untuk dikerjakan secara manual berhubungan karena algoritmanya yang cukup panjang dan kompleks.

Inti dari metode Word Auto Key Encryption (WAKE) terletek pada proses pembentukan tabel S-Box dan proses pembentukan kunci. Tabel S-Box dari metode Wake bersifat fleksibel dan berbeda-beda untuk setiap putran. Secara keseluruhan proses dalam algoritma wake adalah sebagai berikut :

1. Proses pembentukan tabel S-Box
2. Pembentukan kunci
3. Proses enkripsi dan dekripsi

Berikut dibawah ini langkah-langkah di dalam proses pembentukan table S-box

1. Inisialisasi nilai TT[0] … TT[7] :

   TT[0] : 726A8F3B (dalam heksadesimal)

   TT[1] : E69A3B5C (dalam heksadesimal)

   TT[2] : D3C71FE5 (dalam heksadesimal)

   TT[3] : AB3C73D2 (dalam heksadesimal)

   TT[4] : 4D3A8EB3 (dalam heksadesimal)

   TT[5] : 0396D6E8 (dalam heksadesimal)

   TT[6] : 3D4C2F7A (dalam heksadesimal)

TT[7] : 9EE27CF3 (dalam heksadesimal)

2. Inisialisasi untuk nilai awal T[0]…T[3] :

   T[0] = K[0]

   T[1] = K[1]

   T[2] = K[2]

   T[3] = K[3]

   K[0], K[1], K[2], K[3] merupakan inputan kunci yang dipecah menjadi empat bagian.

3. Untuk T[4] sampai T[255], lakukan proses berikut :

   X = T[n-4] + T[n-1]

   T[n] = X >> 3 XOR TT(X AND 7)

4. Untuk T[0] sampai T[22], lakukan proses berikut : T[n] = T[n] + T[n+89]

5. Set nilai untuk beberapa variabel di bawah ini :

   X = T[33]

   Z = T[59] OR (01000001h)

   Z = Z AND (FF7FFFFFh)

   X = (X AND FF7FFFFFh) + Z

6. Untuk T[0] … T[255], lakukan proses berikut :

   X = (X AND FF7FFFFFh) + Z

   T[n] = T[n] AND 00FFFFFFh XOR X

7. Inisialisasi nilai untuk beberapa variabel berikut ini :

   T[256] = T[0]

   X = X AND 255

8. Untuk T[0] … T[255], lakukan proses berikut : Temp = (T[n XOR X] XOR X) AND 255

   T[n] = T[Temp]

   T[X] = T[n+1]

Proses pembentukan kunci dari metode wake dapat ditentukan sendiri yaitu sebanyak n putaran. Semakain banyak putaran dari proses pembentukan kunci, maka keamanan datanya akan semakin terjamin. Fungsi yng digunakan dalam proses pembentukan kunci adalah M(X,Y) = (X+Y) >> 8 XOR T[(X + Y) AND 225]. Pertamatama kunci yang di input akan dipecah menjadi 4 bagian dan di set sebagian awal dari variabel A_0, B_0, C_0, Dan D_0. Nilai dari variabel ini akan di proses dengan melalui langkah-langkah berikut.

A_{i+1} = M(A_i, D_i)

Bi+1 = M(Bi, Ai+1)

Ci+1 = M(Ci, Bi+1)

Di+1 = M(Di, Ci+1)

Nili dari D_i merupakan milai dari kunci K_i.

Inti dari metode WAKE tidak terletak pada proses enkripsi dan dekripsinya, karena proses enkripsi dan dekripsinya hanya berupa XOR dari *plaintext* dan kunci untuk menghasilkan *chipertext* atau operasi XOR *chipertext* dan kunci untuk menghasilkan *plaintext* [1]. Adapun rumus untuk mendapat hasil dari enkripsi dan dekripsi adalah sebagai berikut:

P = C □ K

C = P □ K Dimana :

P = *Plaintext*

K = Kunci

C = *Chipertext*

II.2.6. Keamanan

Masalah keamanan merupakan salah satu aspek terpenting pada sebuah sistem informasi. Masalah keamanan sering kali kurang mendapatkan perhatian dari para perancang dan pengolahan sistem informasi serta berada di urutan setelah tampilan, atau bahkan diurutan terakhir dalam daftar yang dianggap penting.Apabila menggagu peforma sistem, sering kali masalah keamanan tidak begitu dipedulikan bahkan ditiadakan.

Keamanan data adalah kebebasan dari bahaya atau sebagai kondisi keselamatan.Keamanan secara rinci adalah perlindungan data di dalam sesuatu sistem melawan terhadap otorisasi tidak sah, modifikasi, atau perusakan dan perlindungan sistem komputer terhadap pengguna yang tidak bertanggung jawab.

II.2.7. Data Siswa

Data siswa adalah orang yang berlajar di pendidikan (SD, SMP, SMA) dan didalam data siswa terdapat NIM/NIS, nama, tempat, tanggal lahir, alamat, agama, media, no telepon, dan nilai-nilai siswa, prestasi yang diraih nama orang tua dan wali. Sebagai seorang kepala sekolah/guru kita perlu mengetahui data siswa karena

data tersebut penting. Melalui data tersebut kita dapat mengetahui masing-masing nilai siswa dan perbedaan individu/siswa.

II.2.8. Microsoft Visual Basic

Visual Basic (VB) adalah sebuah alat untuk mengembangkan dan membangun aplikasi yang bergerak di atas sistem NET.Fremework, dengan menggunakan bahasa basic. Dengan menggunakan alat ini. Para *programmer* dapat membangun aplikasi Windows forms, aplikasi web berbasis ASP.NET, dan juga aplikasi *command line*. Alat ini dapat diperoleh secara terpisah dari beberapa produk lainnya (Seperti Microsoft Visual C++, Visual C#, atau Visual J#), atau juga dapat diperoleh secara terpadu dalam microsoft visual studio.NET. Bahasa visual Basic .Net sendiri menganut paradigm bahasa pemrograman berorientasi objek yang dapat dilihat sebagai evolusi dari Microsoft visual Basic versi sebelumnya yang diimplementasikan di atas NET Framework. Peluncurannya mengundang kontroversi, mengingat banyak sekali perubahan yang dilakukan oleh Microsoft, dan versi baru ini tidak kompatibel dengan versi terdahulu. (Yonanta Ferdinan Susanto, 2018 : 35)

II.2.9. SQL Server

SQL adalah adalah singkatan dari Structured Query Language. SQL server adalah sistem manajemen database relasional (RDBMS) yang dirancang untuk aplikasi dengan arsitektur client/server. Istilah client, server, dan Client/Server dapat digunakan untuk merujuk kepada konsep yang sangat umum atau hal yang spesifik dari perangkat keras atau perangkat lunak, pada level yang sangat umum,

client adalah setiap komponen dari sebuat sistem yang meminta layanan atau sumber daya (resource) dari komponen sistem lainnya. Server adalah setiap komponen sistem yang menyediakan layanan atau sumber daya ke komponen sistem lainnya. (Yonanta Ferdinan Susanto, 2018 : 35).

II.2.10. UML (Unified Modeling Language)

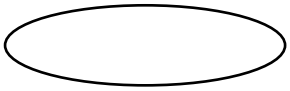
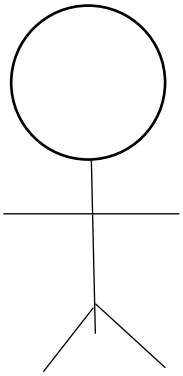
UML Merupakan Metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak dipergunakan dalam industry perangkat lunak dan pengembangan sistem (Urva dan Siregar, 2015 : 93).

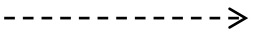
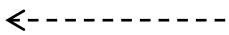
Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut :

1. *Use Case* Diagram

   Use Case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use Case mendekripsikan sebuah interaksi anatara satu atau lebih actor dengan sistem informasi yang akan dibuat. Dapat dikatakan use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam use case diagram dapat dilihat pada tabel II.1. dibawah ini.

Tabel II.1. Simbol *Use Case*

Gambar	Keterangan
	*Use case* menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan actor, dan dinyatakan dengan menggunakan kata kerja awal nama *use case*
	Aktor adalah abstraction dari orang atau sistem yang lain mengktikan fungsi dari target sistem untuk mengidentifikasikan actor, harus ditentukan pembagian tenaga kerja dn tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang tau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan use case, tetapi tidak memiliki control terhadap use case.
	Asosiasi antara actor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.

	Asosiasi antara actor dan use case yang menggunakan panah terbuka untuk mengindinkasikan bila actor berintakasi secara pasif dengan sistem.
	*Include*, merupakan di dalam use case lain (Required) atau pemanggilan use case oleh use case lain, contohnya adalah pemanggilan sebuah fungsi program.
	*Extend*, merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.

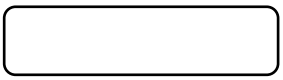
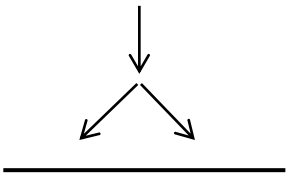
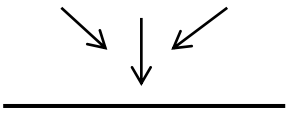
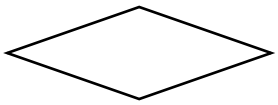
(Sumber : Urva dan Siregar, 2015 : 94)

2. Diagram Aktivitas (*Activity Diagram*)

Activity Diagram menggambarkan Workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktifitas menggambarkan aktifitas sistem bukan apa yang dilakukan actor, jadi aktifitas yang dapat dilakukan oleh sistem (Urva dan siregar, 2015 : 94). Simbol-simbol yang digunakan dalam *Activity diagram* dapat dilihat pada tabel II.2. dibawah ini.

Tabel II.2. Simbol *Activity Diagram*

Gambar	Keterangan
	*Start point*, diletakan pada pojok kiri atas dan merupakan awal aktivitas.

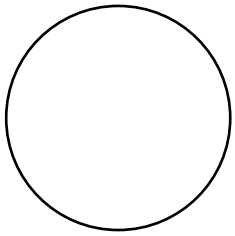
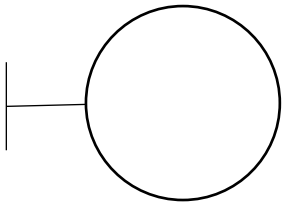
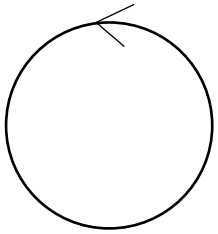
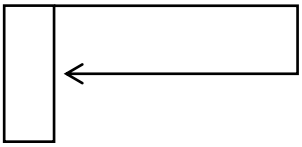
	*End point*, akhir aktifitas .
	*Activites*, menggambarkan suatu proses/kegiatan bisnis.
	*Fork* (percabagan), digunakan untuk menunjukkan kegiatan yang dilakaukan secara parallel atau untuk menggabungan dua kegiatan pararel menjadi satu.
	*Join* (Penggabungan) atau rake, digunakan untuk menunjukkan adanya dekomposisi.
	*Decision points*, menggambarkan pilihan untuk pengambilan keputusan, true, false,
	*Swimlane*, untuk menunjukan siapa melakukan apa.

(Sumber : Urva dan Siregal, 2015 : 94)

3. Diagram Urutan (*Sequence Diagram*)

Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antara objek. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang dilibatkan dalam sebuah use case beserta metode-metode yang dimiliki kelas yang di instasikan menjadi objek itu (Urva dan Siregar, 2015 : 95). Simbol-simbol yang digunakan dalam *Sequence diagram* dapat dilihat pada tabel II.3. dibawah ini.

Tabel II.2. Simbol *Sequence Diagram*

Gambar	Keterangan
	*Entity Class*, merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	*Boundary Class*, berisi kumpulan kelas yang menjadi interfase atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan formentry dan form cetak.
	*Control Class*, atau objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berabagai objek.
	*Message*, simbol mengirim pesan antar class.
	*Recursive*, menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	Activation, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.

	Lifeline, garis titik-titik yang berhubung dengan objek, sepanjang lifeline terdapat activation.

(Sumber : Urva dan Siregar, 2015 : 95)

4. Class Diagram (Diagram Kelas)

Merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem, juga memperhatikan aturan-aturan dan tanggung jawab entitas yang menentukan perilakukan sistem. Class diagram juga menunjukan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang dikoneksikan. Class diagram secara khas meliputi : Kelas (Class), Relasi, Assoclastions, Generalization dan Aggregation, Atribut (Attrabutes), Operasi (Operasiton/Method), Visibility, tingkat akses objek eksternal kepada suatu operasi atau atribut (Urva dan Siregar, 20215 : 95). Hubungan antar kelas mempunyai keterangan yang disebut dengan Multiplicity atau kardinaliti yang dapat dilihat pada tabel II.4. dibawah ini.

Tabel II.2. *Multiplicity Class Diagram*

Multiplicty	Keterangan
1	Satu dan hanya satu
0..*	Boleh tidak ada, satu atau lebih
1..*	Satu atau Lebih
0..1	Boleh tidak ada, maksimal satu

n..n	Batasan antara contoh 2.4 mempunyai arti minimal 2 maksimum 4.

(Sumber : Urva dan Siregal, 2015 : 95)