BAB II

TINJAUAN PUSTAKA

2.1. Kriptografi

Dewasa ini, perkembangan teknologi yang maju membawa dampak pada hamper seluruh aspek kehidupan manusia, seperti dalam hal komunikasi dengan orang lain. Media-media komunikasi bermunculan, seperti telepon genggam dan sampai ke internet yang dapat menghubungkan kita dengan setiap orang dimanapun dia berada. Namun, lalu lintas informasi yang beredar baik dalam media telepon ataupun internet tidaklah terjamin keamanannya. media komunikasi umum yang dapat digunakan oleh siapapun sehingga sangat rawan terhadap penyadapan informasi oleh pihak-pihak yang tidak berhak mengetahui informasi tersebut. Oleh karena penggunaan internet yang sangat luas seperti pada bisnis, perdagangan, bank, industri dan pemerintahan yang umumnya mengandung informasi yang bersifat rahasia maka keamanan informasi menjadi faktor utama yang harus dipenuhi. Berbagai hal telah dilakukan untuk mendapatkan jaminan keamanan informasi rahasia ini.

Untuk itulah diperlukan ilmu kriptografi, yang mempelajari teknik-teknik menyandikan suatu pesan dengan algoritma-algoritma tertentu. Pada dasarnya ada dua metode algoritma, yaitu algoritma rahasia dan algoritma kunci. Metode algoritma rahasia adalah algoritma yang pertama kali dibuat, akan tetapi metode ini tidak efisien untuk digunakan dalam berkomunikasi. Sedangkan, metode algoritma kunci diciptakan setelah penggunaan metode algoritma rahasia yang dirasa tidak efisien untuk digunakan lagi. Metode ini tidak menumpukan keamanan pada algoritmanya, tetapi pada kerahasiaan kunci yang digunakan pada proses penyandian. Metode algoritma kunci mempunyai tingkat efisiensi dan keamanan yang lebih baik dibandingkan dengan algoritma rahasia. Sampai sekarang algoritma kunci masih digunakan secara luas di internet dan terus dikembangkan untuk mendapatkan keamanan yang lebih baik[8].

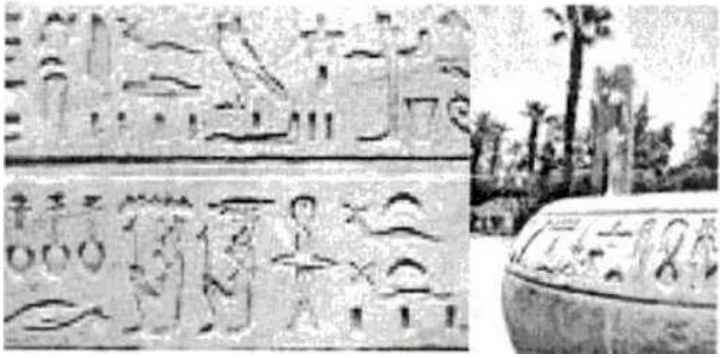
2.2. Sejarah kriptografi

Kriptografi *(cryptography)* berasal dari Bahasa Yunani: "*cryptós*" artinya "*secret*" (rahasia), sedangkan *"gráphein"* artinya *"writing"* (tulisan). Jadi, kriptografi berarti *"secret writing"* (tulisan rahasia). Ada beberapa definisi kriptografi yang telah dikemukakan didalam berbagai literatur. Definisi yang dipakai di dalam buku-buku yang lama (sebelum tahun 1980-an) menyatakan bahwa kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Kriptografi adalah ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, serta otentikasi. Kata "seni" didalam definisi diatas berasal dari fakta sejarah bahwa pada masa-masa awal sejarah kriptografi, setiap orang mungkin mempunyai cara yang unik untuk merahasiakan pesan. Cara-cara unik tersebut 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 menyiratkan sebuah seni) [9].

Kriptografi mempunyai sejarah yang sangat menarik dan panjang. Kriptografi sudah digunakan 4000 tahun yang lalu, diperkenalkan oleh orang-orang Mesir lewat *hieroglyph.* Kode ini adalah rahasia yang hanya dapat diketahui oleh ahli-ahli yang digunakan untuk mengirimkan pesan atas nama raja-raja. Jenis tulisan ini bukanlah bentuk standar untuk menulis pesan.

Dikisahkan pada zaman romawi kuno, pada saat Julius caesar ingin mengirimkan pesan rahasia kepada seorang jendral di medan perang. Pesan tersebut harus dikirimkan melalui seorang kurir. Karena pesan tersebut sangat rahasia, julius tidak ingin pesan tersebut sampai diketahui saat di jalan. Julius caesar kemudian mengacak pesan tersebut sehingga menjadi suatu pesan yang tidak dapat dipahami oleh siapapun terkecuali jendralnya saja. Sebelumnya, Jenderal telah diberi tahu cara membaca pesan yang telah teracak tersebut. Yang

dilakukan Julius Caesar dengan cara mengganti semua susunan alfabet dari a, b, c yaitu a menjadi b, b menjadi c dan c menjadi d dan seterusnya sampai kalimat tersebut tidak bisa dibaca siapapun [10].



Gambar 2.1 Tulisan yang menggunakan *hieroglyph*

[10].

Secara historis terdapat 4 kelompok orang yang berkontribusi terhadap perkembangan kriptografi, dalam komunikasi pesan penting, mereka menggunakan kriptografi untuk menjamin kerahasiaan yaitu kalangan militer (termasuk intelijen dan mata-mata), kalangan diplomatik, penulis buku harian, dan pencinta (*lover*). Di antara keempat kelompok ini, kalangan militer yang memberikan kontribusi paling penting karena pengiriman pesan di dalam suasana perang dibutuhkan teknik enkripsi dan dekripsi yang sangat rumit [9].

Tujuan kriptografi secara umum yaitu mewujudkan keempat aspek keamanan tersebut dalam teori dan praktek yaitu :

1.Kerahasiaan *(confidentiality)*

adalah aspek yang berhubungan dengan penjagaan isi informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah dienkripsi.

2.Integritas data *(data integrity)*

adalah aspek yang berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.

3.Otentikasi *(authentication)*

adalah aspek yang berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain.

4.Penyangkalan *(Non-repudiation)*

Adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman suatu informasi oleh yang mengirimkan, atau harus dapat membuktikan bahwa suatu pesan berasal dari seseorang, apabila ia menyangkal mengirim informasi tersebut[8].

2.3. Istilah Pada Kriptografi

Di dalam kriptografi kita akan sering menemukan berbagai istilah. Istilah yang penting untuk diketahui ialah sebagai berikut.:

a. 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 berupa data atau informasi yang dikirim (melalui kurir, saluran telekomunikasi, dsb) atau yang disimpan di dalam media perekaman (kertas, storage, dsb). Pesan yang tersimpan tidak hanya berupa teks, tetapi juga dapat 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 bentuk lain yang tidak dapat dipahami. Bentuk pesan yang tersandi disebut cipherteks *(ciphertext)* atau kriptogram *(cryptogram)*. Cipherteks harus dapat ditransformasikan kembali menjadi plainteks semula agar pesan yang diterima bisa dibaca. Plainteks (*plaintext* atau *cleartext*, artinya teks asli): pesan yang ingin dirahasiakan. Pesan dapat berupa data atau informasi yang dikirim atau yang disimpan di dalam media perekaman yang dapat disimpan dalam kertas, storage (*disk*, kaset, *CD*) dan sebagainya.

b. 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 berupa orang, mesin (komputer), kartu kredit, dan sebagainya. Jadi, orang bisa bertukar pesan dengan orang lainnya (contoh: Alice berkomunikasi dengan Bob), 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.

c. Enkripsi dan dekripsi

Proses menyandikan plainteks menjadi cipherteks disebut enkripsi *(encryption)* atau enciphering (standard nama menurut ISO 7498-2). Sedangkan proses mengembalikan cipherteks menjadi plainteks semula dinamakan dekripsi *(decryption)* atau deciphering (standard nama menurut ISO 7498-2). Enkripsi dan dekripsi dapat diterapkan baika pada pesan yang dikirim maupun pada pesan tersimpan. Istilah encryption of data in motion mengacu pada enkripsi pesan yang ditransmisikan melalui saluran komunikasi, sedangkan istilah enrypton of data at-rest mengacu pada enkripsi dokumen yang disimpan di dalam storage. Contoh encryption of data in motion adalah pengiriman nomor PIN dari mesin ATM ke komputer *server* di kantor bank pusat.

d. Cipher

Algoritma kriptografi disebut juga cipher yaitu aturan untuk *enchipering* dan dechipering, atau fungsi matematika yang digunakan 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 elemenelemen antara kedua himpunan tersebut. Misalkan P menyatakan plainteks dan dan C menyatakan chiperteks, maka fungsi enkripsi E memetakan P ke C,

$$E(P) = C$$

Dan fungsi dekripsi D memetakan C ke P,

$$D(C) = P$$

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

$$D(E(P)) = P$$

Keamanan algoritma sering kriptografi diukur dari banyaknya kerja (work) yang dibutuhkan untuk memecahkan chiperteks menjadi plainteksnya tanpa mengetahui kunci yang digunakan. Kerja ini dapat diekivalenkan dengan waktu, memori, uang, dan lain-lain. Semakin banyak kerja yang diperlukan, yang berarti juga semakin lama waktu yang dibutuhkan, maka semakin kuat algoritma kriptografi tersebut, yang berarti semakin aman digunakan untuk menyandikan pesan. Jika keamanan kriptografi ditentukan dengan menjaga kerahasiaan algoritmanya, maka algoritma kriptografinya dinamakan algoritma *restricted*. Algoritma *restricted* mempunyai sejarah tersendiri di dalam kriptografi. Algoritma *restricted* biasanya digunakan oleh sekelompok orang 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 cocok lagi saat ini, sebab setiap kali ada anggota kelompok keluar, maka algoritma kriptografi harus diganti lagi. Kriptografi modern mengatasi masalah di atas dengan

penggunaan kunci, yang dalam hal ini algoritma tidak lagi dirahasiakan, tetapi kunci harus d jaga kerahasiaannya.

e. kunci

parameter yang digunakan untuk transformasi *enciphering* dan *dechipering*. Kunci biasanya berupa string atau deretan bilangan.

f. Sistem Kriptografi

Kriptografi membentuk sebuah sistem yang dinamakan sistem kriptografi. Sistem kriptografi *(cryptosystem)* adalah kumpulan yang terdiri dari algoritma kriptografi, semua plainteks dan cipherteks yang mungkin, dan kunci [SCH96]. Di dalam sistem kriptografi, cipher hanyalah salah satu komponen saja.

g. Penyadap

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

h. Kriptanalisis dan kriptologi

Kriptografi berkembang sedemikian rupa sehingga melahirkan bidang yang berlawanan yaitu kriptanalisis. Kriptanalisis *(cryptanalysis)* adalah ilmu dan seni untuk memecahkan chiperteks menjadi plainteks tanpa mengetahui kunci yang digunakan. 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 kriptografi maupun kriptanalisis keduanya saling berkaitan.

Gambar 2.2 Proses Enkripsi dan Dekripsi

2.4 Macam-macam Algoritma Kriptografi

Algoritma kriptografi dibagi menjadi tiga bagian dengan berdasarkan kunci yang dipakainya [10] :

1.Algoritma Simetrik : menggunakan satu kunci untuk proses enkripsi dan dekripsinya.
2. Algoritma Asimetrik : menggunakan kunci yang berbeda untuk proses enkripsi dan dekripsinya.

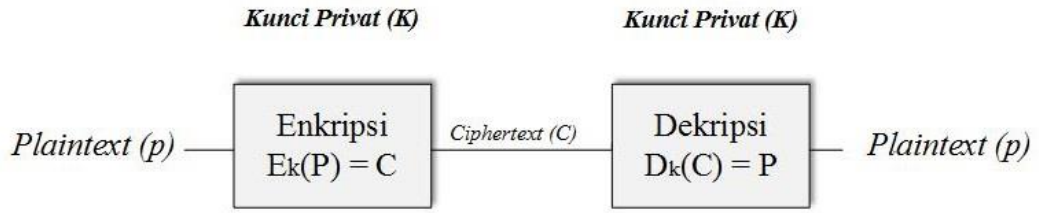
2.4.1 Algoritma Simetrik

Algoritma simetrik ini sering disebut dengan Algoritma klasik karena memakai kunci yang sama untuk proses enkripsi dan dekripsi. Algoritma klasik sudah ada sejak lebih dari 4000 tahun yang lalu. Algoritma simetrik ini adalah algoritma yang paling sering atau umum digunakan. Jika mengirim pesan menggunakan algoritma klasik pada sisi penerima pesan harus terlebih dahulu diberitahu kunci dari pesan tersebut agar penerima dapat mendekripsikan atau membaca pesan yang telah dikirim. Keamanan pesan yang menggunakan algoritma klasik ini tergantung pada kuncinya. Apabila kunci tersebut diketahui oleh orang yang tidak berhak mengakses pesan tersebut maka orang tersebut dapat melakukan proses enkripsi dan dekripsi terhadap pesan. B Proses enkripsi dan dekripsi dari fungsi algoritma ini dapat dinotasikan sebagai berikut :

Ek (P) = C (Proses Enkripsi)

Dk (C) = P (Proses Dekripsi)

Dimana E disini merupakan fungsi enkripsi, D ialah fungsi dekripsi, k adalah kunci enkripsi dan dekripsi, P adalah *plaintext* (pesan asli) dan C adalah *ciphertext* (hasil proses enkripsi ke *plaintext*). Pada proses enkripsi dan dekripsi dengan menggunakan algoritma simetrik dapat digambarkan sebagai berikut :



Gambar 2.3 Proses Enkripsi dan Dekripsi Algoritma Simetrik

Macam-macam algoritma yang menggunakan kunci simetrik ialah, Data Encryption Standart Cipher Permutasi, DES dan IDEA , *vigenere cipher*, Cipher Substitusi, Cipher Hill, OTP, RC6, Twofish, FEAL, SAFER, LOKI, CAST, Rijndael (AES), Blowfish, GOST, A5 [10].

2.4.1.1 Kelebihan dan kekurangan kunci simetrik

Kelebihan dan kekurangan dari Algoritma kunci simetrik ini ialah

Kelebihan kunci simetrik :

a.Kecepatan operasi lebih tinggi bila dibandingkan dengan algoritma asimetrik.

b.Karena kecepatannya yang cukup tinggi, sehingga algoritma ini bisa digunakan untuk sistem *real-time*

c.Pada pendistribusian kunci dapat lebih aman.

Kelemahan kunci simetrik:

a.Proses pengiriman pesan dengan pengguna yang berbeda dibutuhkan kunci yang berbeda juga, sehingga akan tejadi kesulitan pada manajemen kunci.

b.Permasalahan dalam pengiriman kunci itu sendiri yang disebut *"key distribution problem"* .

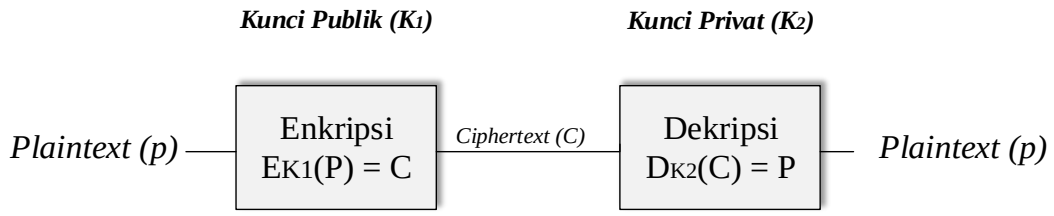
2.4.2 Algoritma Asimetrik

Algoritma asimetrik yang sering juga disebut dengan algortima kunci publik (*public key*), yang artinya kunci yang digunakan pada proses enkripsi dan dekripsi berbeda. Pada algoritma asimetri kunci terbagi menjadi dua bagian yaitu :

1. Kunci umum *(Public key)* : kunci yang semua orang tahu (dipublikasikan)
2. Kunci rahasia (*Private key*) : kunci yang dirahasiakan (hanya boleh diketahui oleh satu orang).

Kunci publik bersifat umum, artinya kunci ini tidak dirahasiakan sehingga dapat dilihat oleh siapa saja. Sedangkan kunci rahasia adalah kunci yang dirahasiakan dan hanya orang-orang tertentu saja yang boleh mengetahuinya. Keuntungan utama dari algoritma ini adalah memberikan jaminan keamanan kepada siapa saja yang melakukan pertukaran informasi meskipun di antara mereka tidak ada kesepakatan mengenai keamanan pesan terlebih dahulu maupun saling tidak mengenal satu sama lainnya[11].

Kriptografi kunci-publik dapat dianalogikan seperti kotak surat yang terkunci dan memiliki lubang untuk memasukkan surat. Setiap orang dapat memasukkan surat ke dalam kotak tersebut, tetapi hanya pemilik kotak yang dapat membuka kotak dan membaca surat di dalamnya karena ia yang memiliki kuncinya. Keuntungan kriptografi kunci-publik ada dua. Pertama, kunci publik dapat dikirim ke penerima melalui saluran yang sama dengan saluran yang digunakan untuk mengirim pesan (saluran untuk mengirim pesan umunya tidak aman). Kedua, jumlah kunci dapat ditekan. Untuk berkomunikasi dengan banyak orang tidak perlu kunci rahasia sebanyak orang tersebut, cukup membuat dua kunci, yaitu kunci publik dan kunci rahasia[12].



Gambar 2.4 Proses Enkripsi dan Dekripsi Algoritma Asimetrik

2.4.2.1 Kelebihan dan Kekurangan kunci Asimetrik

Kelebihan dan kelemahan dari algoritma simetrik ini ialah

Kelebihan :

a. Pada pendistribusi keamanan kunci dapat lebih baik.
b. Manajemen kunci yang lebih baik dikarena jumlah kunci yang lebih sedikit.

Kelemahan :

a. Untuk kecepatan nya lebih rendah dibandingkan dengan algoritma simetrik.
b. Tingkat keamanan sama, kunci yang digunakan lebih panjang dibandingkan dengan algoritma simetris [10].

2.5 Kriptografi Klasik

Kriptografi klasik adalah suatu algoritma yang pada pengamanan pesan hanya membutuhkan satu kunci. Teknik ini telah digunakan beberapa abad yang lalu. Dua teknik dasar yang biasa digunakan pada algoritma jenis ini adalah sebagai berikut:

1. Teknik substitusi: Penggantian setiap karakter teks-asli dengan karakter lain.
2. Teknik transposisi (permutasi): Dilakukan dengan menggunakan permutasi karakter [10].

2.6 Algoritma *Hybrid*

Pada Algoritma *hybrid* melalui proses penggabungan antara kriptografi simetrik dan asimetrik. Metode kriptografi dibagi menjadi kriptografi simetrik dan asimetrik. Perbedaan dari kedua algoritma tersebut terdapat pada kunci yang digunakan untuk proses enkripsi dan dekripsi nya. Karena setiap algoritma mempunyai keamanan dan kelemahan nya masing-masing, maka untuk pengamanan yang maksimal di padukan antara dua metode kriptografi yaitu perpaduan antara algoritma simetrik dan asimetrik. Perpaduan kedua metode ini disebut kriptografi *hybird*. Pada pengamanan menggunakan kriptografi simetris lebih efisien dalam proses enkripsi dan dekripsi namun kriptografi simetris ini memiliki kelemahan pada proses pendistribusian kuncinya. Namun untuk kriptografi asimetris kelemahan nya ialah kurang efisien saat proses enkripsi dan dekripsi dikarenakan membutuhkan waktu yang lebih lama bila dibandingkan pada kriptografi simetris. Tetapi pada kriptografi asimetris mempunyai keuntungan yaitu tidak memerlukan proses distribusi kunci karena kunci yang akan digunakan pada proses enkripsi dan dekripsi ditempatkan pada jaringan umum [7].

2.6.1 Proses Kriptografi *Hybrid*

Proses kriptografi *hybrid* yaitu yang pertama melakukan proses enkripsi pada *plainteks* agar dapat memperkuat pertahanan terhadap serangan penyerang (*attacker*) yang pada umumnya memanfaatkan pola-pola yang ada pada *plainteks* untuk memecahkan *chiperteks* menggunakan algoritma simetris. Setelah *plainteks* menjadi teks terenkripsi kemudian kunci privat algoritma simetris dienkripsi menggunakan pasangan kunci publik algoritma asimetris kemudian dikirimkan. Untuk pengenkripsian kunci ini disebut *session key* yaitu kunci privat yang terenkripsi yang bersifat tercipta hanya pada saat itu juga (*one-time only*). Kunci yang terenkripsi bersama dengan chiperteks kemudian ditransmisikan kepada penerima. Untuk proses dekripsi, penerima menerima paket tersebut

menggunakan pasangan kunci privat algoritma asimetris untuk mendekripsi *session key* terlebih dahulu. Lalu dengan *session key* tersebut, kunci privat algoritma simetris dapat dibuka dan penerima dapat mendekripsi *chipertext* tersebut menjadi *plaintext* kembali [7]. Algoritma yang akan digunakan untuk pengamanan data ini adalah algoritma *Vigenere Cipher* dan RSA dimana metode ini menggabungkan kelebihan masing-masing algoritma tersebut.

2.7 *Vigenere cipher*

Vigenere termasuk dalam kode abjad-majemuk (*polyalphabetic substitution cipher*). Dipublikasikan oleh diplomat (sekaligus seorang kriptologis) Perancis yang bernama Blaise de Vigenere pada abad ke 16, tahun 1586. Sebenarnya pada tahun 1553, Giovan Batista Belaso telah menggambarkannya untuk pertama kali seperti yang ditulis di dalam buku *La Cifra del Sig*. Setelah 200 tahun kemudian, algoritma tersebut baru dikenal luas dengan dinamakan kode *vigenere*. *Vigenere* merupakan pemicu bagi perang sipil di Amerika dan kode *Vigenere* juga digunakan oleh tentara konfiderasi (*confederate army*) pada perang sipil Amerika (*American civil war*). Pada pertengahan abad ke 19, kode *vigenere* berhasil dipecahkan oleh *Babbage* dan *Kasiski*.

Bila pada teknik di atas, setiap teks kode harus selalu mengganti nilai setiap teks asli tertentu (tidak peduli apakah jumlah teks kode tersebut ekuivalen dengan teks asli tertentu satu atau lebih). Pada teknik substitusi *vigenere* setiap teks kode bisa mempunyai banyak kemungkinan teks asli. Teknik dari substitusi *vigenere* dapat dilakukan dengan dua cara, yaitu angka dan huruf [10].

2.8 *RSA (Rivest-Shamir-Adleman)*

RSA termasuk jenis algoritma asimetris. Dari sekian banyak algoritma kriptografi yang telah dibuat, algoritma ini lah yang paling populer dan banyak orang menggunakan nya untuk proses pengamanan pesan. Karena algoritma tersebut dibangkitkan dengan menggunakan dua kali proses kunci yaitu kunci *public* yang digunakan untuk proses enkripsi dan kunci *private* yang digunakan untuk proses dekripsi pesan. Dengan menggunakan kunci publik semua orang dapat mengakses pesan tersebut. Namun, pesan yang terbaca ialah berupa *ciphertext* yaitu pesan yang telah diacak. Untuk membaca *plaintext* nya digunakan kunci *private* yaitu hanya orang yang berhak dapat mengakses pesan tersebut. Algoritma ini dibuat oleh 3 orang peneliti dari MIT (*Massachussets Institute of Technology*) pada tahun 1976. Nama algoritma kriptografi RSA ini diambil dari singkatan nama penemu yaitu Ron Rivest, Adi Shamir dan Leonard Adleman. Kekuatan algoritma RSA terletak pada tingkat kesulitan dalam memfaktorkan bilangan menjadi faktor primanya, yang dalam hal ini adalah memfaktorkan n menjadi a dan b. Sekali n berhasil difaktorkan menjadi p dan q, maka m= (p - 1)(q - 1) dapat dihitung. Selanjutnya karena kunci enkripsi diutamakan e diumumkan (tidak rahasia), maka kunci deskripsi d dapat dihitung dari persamaan ed = 1 (mod m). Penemu algoritma RSA menyarankan nilai p dan q panjangnya lebih dari 100 digit. Dengan demikian hasil kali n= pq akan berukuran lebih dari 200 digit. Bayangkanlah berapa besar usaha kerja yang diperlukan untuk memfaktorkan bilangan bulat 200 digit menjadi factor primanya. Menurut Rivest dan kawan-kawan untuk mencari faktor bilangan 200 digit membutuhkan waktu komputasi selama 4 milyar tahun (dengan asumsi bahwa algoritma pemfaktoran yang digunakan adalah algoritma yang tercepat saat ini dan komputer yang dipakai mempunyai kecepatan 1 mil detik[14].

2.9 Serangan Pada Kriptografi

Ada banyak faktor yang dapat mengancam keamanan data. Ancaman-ancaman tersebut menjadi masalah dimana dengan semakin meningkatnya komunikasi data yang bersifat rahasia misalnya pemindahan data secara elektronik pada dunia perbankan atau pengiriman dokumen rahasia pada instansi pemerintah. Untuk mengantisipasi berbagai ancaman-ancaman yang ada perlu dilakukan usaha untuk melindungi data yaitu salah satu nya teknik enkripsi.

Serangan sistem kriptografi pada dasarnya dapat dibedakan menjadi dua jenis yaitu:

1.Serangan aktif

Pada serangan ini, penyerang mengintervensi komunikasi kemudian ikut mempengaruhi sistem untuk keuntungan pribadinya. Misalnya *attacker* aliran pesan misalnya menghapus sebagian *ciphertext*, kemudian mengubah nya, dan menyisipkan potongan *ciphertext* palsu, membalas kembali pesan lama, mengubah informasi yang tersimpan, dan sebagainya.

2.Serangan pasif

Pada proses ini, penyerang tidak terlibat dalam komunikasi antara pengirim dan penerima pesan, namun penyerang akan menyadap semua pertukaran pesan antara kedua entitas tersebut, tujuan nya adalah agar *attacker* dapat sebanyak mungkin informasi yang digunakan untuk kriptanalis [15].

Berdasarkan ketersediaan data yang ada, serangan terhadap kriptografi dapat diklasifikasikan menjadi (asumsi yang digunakan: kriptanalis mengetahui algoritma kriptografi yang digunakan):

1.*Chipertext-only attack*

Kriptanalis memiliki beberapa cipherteks dari beberapa pesan, semuanya dienkripsi dengan algoritma yang sama.Tugas kriptanalisis adalah menemukan plainteks sebanyak mungkin atau menemukan kunci yang digunakan untuk mengenkripsi pesan.

2.*Known-plaintext attack*

Beberapa pesan yang formatnya terstruktur membuka peluang kepada kriptanalis untuk menerka plainteks dari cipherteks yang bersesuaian.

3.*Chosen-plaintext attack*

Serangan jenis ini lebih hebat daripada known-plaintext attack, karena kriptanalis dapat memilih plainteks tertentu untuk dienkripsikan, yaitu plainteks-plainteks yang lebih mengarahkan penemuan kunci.

4.*Adaptive-chosen-plaintext attack*

Kasus khusus dari jenis serangan nomor 3 di atas. Misalnya, kriptanalis memilih blok plainteks yang besar, lalu dienkripsi, kemudian memilih blok lainnya yang lebih kecil berdasarkan hasil serangan sebelumnya,begitu seterusnya.

5.*Chosen-Ciphertext attack*

Kriptanalis memiliki akses terhadap cipherteks yang didekripsi (misalnya terhadap mesin elektronik yang melakukan dekripsi secara otomatis).

6.*Chosen-text attack*

Gabungan *chosen-plaintext attack* dan *chosen-ciphertext attack.*

7. *Chosen-key attack*

Kriptanalis memiliki pengetahuan mengenai hubungan antara kunci-kunci yang berbeda, dan memilih kunci yang tepat untuk mendekripsi pesan.

8. *Rubber-hose cryptanalysis*

Kriptanalis mengancam, mengirim surat gelap,atau melakukan penyiksaan sampai orang yang memegang kunci memberinya kunci untuk mendekripsi pesan. Mungkin ini cara yang terbaik untuk memecahkan kriptografi [16].

2.10 ASCII (*American Standard Code for Information Interchange*)

Kode Standar Amerika untuk Pertukaran Informasi atau ASCII (American Standard Code for Information Interchange) merupakan suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". ASCII selalu

digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. Nilai hash yang akan dicari dengan fungsi hash dalam algoritma Dice Similarity Ceoficient merupakan representasi dari nilai ASCII (American Standar Code for Information Interchange) yang menempatkan angkanumerik pada karakter, angka, tanda baca dan karakterkarakter lainnya. ASCII menyediakan 256 kode yang dibagi kedalam dua himpunan standar dan diperluas yang masing-masing terdiri dari 128 karakter. Himpunanini merepresentasikan total kombinasi dari 7 atau 8 bit, yang kemudian menjadi angka dari bit dalam 1 byte. ASCII standar menggunakan 7 bit untuk tiap kode dan menghasilkan 128 kode karakter dari 0 sampai 127 (heksadesimal 00H sampai 7FH). Himp nan ASCII yang diperluas menggunakan 8 bit untuk tiap kode dan menghasilkan 128 kode tambahan dari 128 sampai 255 (heksadesimal 80H sampai FFH). Algoritma Dice Similarity Coeficient melakukan perhitungan nilai hash dengan memperlakukan setiap substring sebagai sebuah angka dengan basis tertentu, di mana basis yang digunakan pada umumnya merupakan bilangan prima yang besar. Misalnya, jika substring yang ingin dicari adalah “dia” dan basis yang digunakan adalah 101, nilai hash yang dihasilkan adalah 100 x 102 + 105 x 101 + 97 x 100 = 11147 (nilai ASCII dari ‘d’ adalah 100, ‘i’ adalah 105, dan nilai ASCII dari ‘a’ adalah 97).

Dalam pengkodean kode ASCII memanfaatkan 8 bit. Pada saat ini kode ASCII telah tergantikan oleh kode *UNICODE (Universal Code). UNICODE* dalam pengkodeannya memanfaatkan 16 bit sehingga memungkinkan untuk menyimpan kode-kode lainnya seperti kode bahasa Jepang, Cina, Thailand dan sebagainya. Pada papan *keyboard*, aktifkan *numlock*, tekan tombol ALT secara bersamaan dengan kode karakter maka akan dihasilkan karakter tertentu. Misalnya: ALT + 44 maka akan muncul karakter koma (,). Mengetahui kode-kode ASCII sangat bermanfaat misalnya untuk membuat karakter-karakter tertentu yang tidak ada di *keyboard* [17]

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	α	240	≡			
129	ü	145	æ	161	í	177	▒	193	┴	209	╤	225	ß	241	±			
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥			
131	â	147	ô	163	ú	179	|	195	├	211	╙	227	π	243	≤			
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠			
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡			
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	µ	246	÷			
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈			
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°			
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙			
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·			
139	ï	155	¢	171	½	187	╗	203	╦	219	█	235	δ	251	√			
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n			
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²			
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	ε	254	■			
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255				

Source: www.LookupTables.com

Gambar 2.5 Tabel ASCII 0-255 karakter

[18]

2.11 *PHP (Personal Hypertext Preprocessor)*

PHP (Hypertext Preprocessor), merupakan bahasa pemrograman pada sisi server yang memperbolehkan programmer menyisipkan perintah – perintah perangkat lunak web server (Apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke browser yang me-request-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman web setiap kali tampilan tanggal dibutuhkan. Sesuai dengan fungsinya yang berjalan di sisi server maka PHP adalah bahasa pemrograman yang digunakan untuk membangun teknologi web application. PHP telah menjadi bahasa *scripting* untuk keperluan umum yang pada awalnya hanya digunakan untuk pembangunan web yang menghasilkan halaman web dinamis. Untuk tujuan ini, kode PHP tertanam ke dalam dokumen sumber HTML dan diinterpretasikan oleh server web dengan modul PHP prosesor, yang menghasilkan dokumen halaman web. Sebagai bahasa pemrograman untuk tujuan umum, kode PHP diproses oleh aplikasi penerjemah dalam modus baris - baris perintah modus dan melakukan operasi yang diinginkan sesuai sistem operasi untuk menghasilkan keluaran program dichannel output standar. Hal ini juga dapat berfungsi sebagai aplikasi grafis. PHP tersedia sebagai prosesor untuk server web yang paling modern dan sebagai penerjemah mandiri pada sebagian besar system operasi dan komputer platform. [18].

2.12 *MySQL* (*My Structure Query Language*)

MySQL adalah sebuah implementasi dari sistem manajemen basisdata relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basisdata yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang

memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Kehandalan suatu sistem basisdata (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basisdata transaksional maupun operasi basisdata nontransaksional. Pada modus operasi nontransaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basisdata kompetitor lainnya [18].

2.13 *XAMPP*

XAMPP adalah sebuah software web server apache yang didalamnya sudah tersedia database *server* MySQL dan dapat mendukung pemrograman PHP. XAMPP merupakan software yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows. Keuntungan lainnya adalah cuma menginstal satu kali sudah tersedia Apache Web Server, MySQL Database Server, PHP Support (PHP 4 dan PHP 5) dan beberapa module lainnya. [18].

2.14. Data *Flow* Diagram (DFD)

Data *Flow* Diagram (DFD) merupakan diagram yang digunakan untuk menggambarkan proses-proses yang terjadi pada sistem yang akan dikembangkan [19].

Untuk membaca suatu DFD harus memahami dulu elemen-elemen yang menyusun suatu DFD [19]. Ada empat elemen yang menyusun suatu DFD yaitu:

1.Process (Proses)

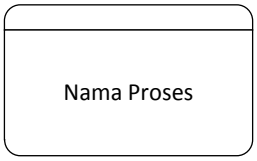
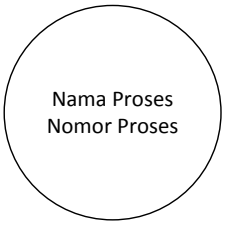
Aktivitas atau fungsi yang dilakukan untuk alas an bisnis yang spesifik, bisa berupa manual maupun terkomputerisasi.

2.Data *Flow* (Arus Data)

Satu data tunggal atau kumpulan logis suatu data, selalu diawali atau berakhir pada proses

3.Data *store* (Simpan Data)

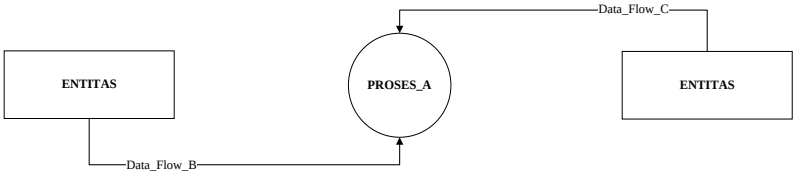
Tabel 2.1 Simbol Elemen-elemen DFD

Elemen Data Flow Diagram	**Field Tipikal yang biasa digunakan**	**Simbol Gane and Sarson**	**Simbol DeMarco and Jourdan**
Setiap proses memiliki: - Nomor - Nama - Deskripsi proses - Satu/lebih output data flow - Satu/lebih input flow	- Label (Nama) - Tipe (Proses) - Deskripsi - Nomor Proses		
Setiap Data *Flow* memiliki: - Nama - Deskripsi - Satu/lebih koneksi ke suatu proses	- Label - Tipe - Deskripsi - Alias - Komposisi (Deskripsi dari elemen-elemen data)		
Setiap Data *Store* memiliki: - Nomor - Nama - Deskripsi - Satu/lebih input data flow - Satu/lebih soutput data flow	- Label (Nama) - Tipe - Deskripsi - Alias - Komposisi - Catatan		
Setiap Entitas memilik: - Nama - Deskripsi	- Label - Tipe - Deskripsi - Alias - Deskripsi Entitas		

[19].

2.14.1. Diagram Konteks

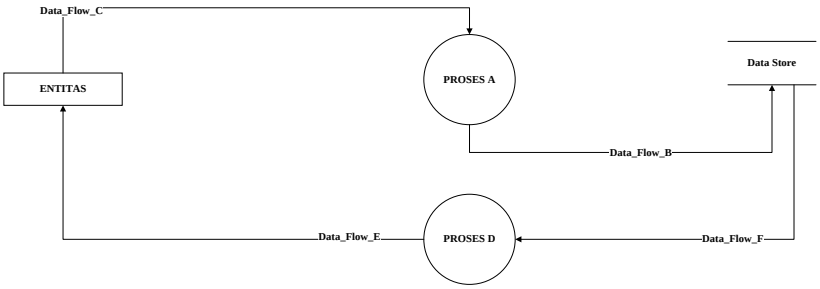
Diagram konteks menunjukkan semua entitas luar yang menerima informasi dari atau memberikan informasi ke sistem [19].



Gambar 2.6 Contoh Bentuk Diagram Konteks

2.14.2. Diagram Nol

Menunjukkan semua proses utama yang menyusun keseluruhan sistem [19].



Gambar 2.7 Contoh Bentuk Diagram Nol

2.15. *Entity Relationship Diagram* (ERD)

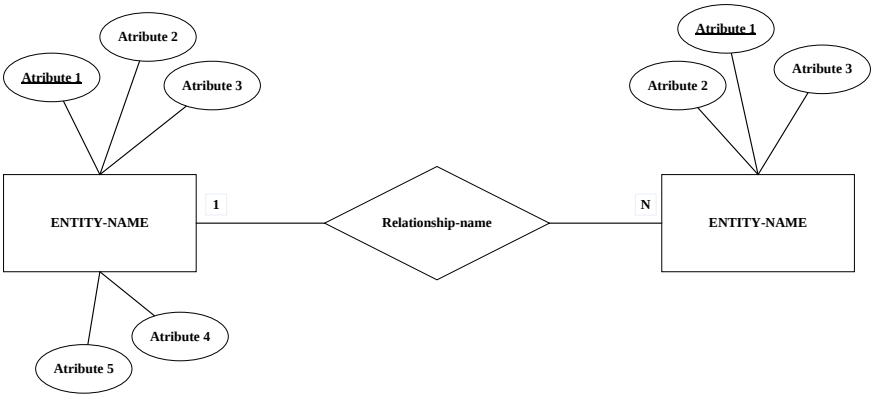
Pengertian *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) adalah gambar atau diagram yang menunjukkan informasi dibuat, disimpan, dan digunakan dalam sistem bisnis [19]. seperti data *flow diagram*, ERD juga menggunakan symbol-simbol khusus untuk menggambarkan elemen-elemen ERD.ERD terbagi dua versi yakni ERD verdi Chen dan versi James Martin. Berikut simbol-simbol yang digunakan dalam masing masing versi seperti yang ditunjukkan oleh tabel 2.2 dan tabel 2.3

Tabel 2.2 Simbol-simbol *Entitas Relationship Diagram*

Keterangan	IDEF1X	Chen
Entitas: Orang, tempat, atau benda memiliki nama tunggal ditulis dengan huruf besar berisi lebih dari 1 *instance*	ENTITY-NAME Identifier	ENTITY-NAME *Identifier
Attribute: Properti dari entitas harus digunakan oleh minimal 1 proses bisnis dipecah dalam detail	ENTITY-NAME Attribute-name Attribute-name	Attribute-name
Relationship: Menunjukkan hubungan antar 2 entitas Dideskripsikan dengan kata kerja memiliki modalitas (*null*/*not null*) Memiliki kardinalitas (:1,1:N, atau M:N)	Relationship-name	Relatio nship-

[19].



[19].

Gambar 2.3 Contoh Bentuk *Entitas Relationship Diagram*

2.16. Perbandingan Metode

Adapun perbandingan antara metode *vigenere cipher* dikombinasikan dengan metode *RSA* yang terdapat pada table berikut.

Tabel 2.1 Perbandingan Metode

	Metode *Vigenere Cipher*	Metode *RSA*	Metode Kombinasi *Vigenere Cipher* dan *RSA*
Pengenalan	Metode yang menggunakan bentuk bujur sangkar *vigenere* atau yang dikenal sebagai *tabula recta*	Metode yang menggunakan dua kali proses pengkuncian yaitu *public key* dan *private key.*	Metode penggabungan antara metode *vigenere cipher* dan RSA
Prinsip Kerja	Menggunakan 26 Karakter	Menggunakan 0-255 kode ASCII	Kedua metode menggunakan 0-255 kode ASCII
Proses	Penyandian *plaintext* menggunakan perpanjangan alfabet berurutan untuk diperpanjang menjadi perpanjangan alfabet berurutan	Penyandian *plaintext* menggunakan tabel ASCII 0-255 karakter.	Penyandian *plaintext* dengan menggunakan metode *vigenere cipher* menggunakan kode ASCII. Proses dan hasil *ciphertext* nya digunakan sebagai *plaintext* pada metode *RSA*