BAB I
PENGENALAN KRIPTOGRAFI

1.1 Pengertian Kriptografi

Kriptografi adalah salah satu cabang ilmu yang berhubungan dengan persandian ataupun kerahasiaan. Dalam bahasa Yunani, kriptografi terdiri dari 2 kata yaitu kryptos artinya “tersembunyi, rahasia” dan graphein artinya “menulis”. Sehingga dapat diartikan sebagai tulisan yang bersifat rahasia.

Apabila 2 kata tersebut digabungkan, bisa diartikan sebagai menulis sesuatu yang tersembunyi. Jadi tulisan tersebut tidak akan diketahui oleh orang lain, khususnya mereka yang tidak paham dengan sandi ataupun rahasianya.

Penggunaannya sudah banyak dilakukan sejak puluhan tahun yang lalu sebagai bentuk dari penulisan suatu rahasia. Bahkan, sudah sangat sering digunakan dalam peperangan agar musuh tidak mengetahui strategi.

Masih tidak paham apa itu kriptografi ?

Kriptografi adalah suatu ilmu yang mempelajari bagaimana cara menyampaikan pesan secara tersembunyi kepada tujuan. Karena pesan yang disampaikan tersembunyi, tidak semua orang bisa mengetahuinya dengan mudah tanpa mengetahui kuncinya.

Kunci dari kriptografi ini akan memudahkan kita semua dalam mengungkap apa isi dari naskah yang sedang dipelajari, ini penting untuk diperhatikan karena kunci biasanya berisi petunjuk atau tutorial langsung untuk membuka makna dari pesan yang sedang dikirimkan.

Kriptografi juga dapat berarti penerapan dan studi tentang teknik pengamanan komunikasi maupun data terhadap pihak ketiga/musuh.

Sebelum menggunakan teknologi yang maju seperti sekarang, para ahli hanya menggunakan cara tradisional. Mereka menerjemahkan dengan berbagai cara dari naskah asli atau plain text, dari sini akan muncul berbagai kemungkinan untuk membuka makna yang ada di dalamnya.

Seiring dengan berjalannya waktu, kriptografi bisa diartikan sebagai teknologi untuk menyimpan informasi dengan ketat. Dengan cara ini, berbagai pesan yang dikirim secara digital akan diacak sehingga hanya pengirim dan penerima saja yang bisa membacanya.

1.1.1 Pengertian Kriptografi Menurut Para Ahli

Terdapat beberapa pengertian kriptografi menurut ahli. Nah berikut ini adalah definisi menurut para ahli.

- Pengertian kriptografi menurut Talbot dan Welsh (2006). Kriptografi adalah sebuah teknik rahasia dalam penulisan, dengan menggunakan karakter khusus, dan menggunakan huruf dan karakter di luar bentuk aslinya ataupun dengan metode-metode yang lain yang hanya bisa dipahami pihak-pihak yang memproses kunci. Jadi secara umum bisa diartikan sebagai seni menulis atam memecahkan cipher.
- Pengertian kriptografi menurut Oppliger (2005), bisa diartikan sebagai sebuah proses untuk melindungi data dalam arti yang luas.
- Pengertian kriptografi menurut Menezes, Oorschot dan Vanstone (1996), yaitu sebuah studi teknik matematika yang berkaitan dengan aspek keamanan informasi seperti kerahasiaan, otentikasi entitas serta otentikasi keaslian data dan integritas data. Tidak hanya penyediaan keamanan informasi saja, tetapi juga sebuah himpunan teknik-teknik.

Kriptografi adalah ilmu yang mempelajari atau berfokus pada berbagai teknik matematika yang berhubungan dengan keamanan informasi (otentikasi dan kerahasiaan). Setelah mengetahui pengertian apa itu kriptografi, mari kita bahas sejarah, tujuan, klasifikasi, algoritma, dan contoh kriptografi.

1.1.2 Istilah-Istilah dalam Kriptografi

Menurut Abidin, dkk (2016), beberapa istilah dalam kriptografi yang sering digunakan adalah sebagai berikut.

- Plaintext [p] = pesan awal atau informasi sebelum dienkripsi.
- Enkripsi [E[p]] = proses kriptografi dari p menjadi c.
- Ciphertext [c] = pesan yang telah dienkripsi, tidak bisa dibaca karena karakter tidak memiliki makna.

- Key [k] = proses dekripsi kunci yang diperoleh penerima.
- Deskripsi [D[c]] = mengubah c ke p.

1.2 Sejarah Kriptografi

Sebelum zaman modern, kriptografi dilihat hanya semata-mata berhubungan dengan pesan rahasia (seperti enkripsi)-konversi pesan dari bentuk dapat dipahami menjadi bentuk yang tak dapat dipahami dan kembali lagi satu dengan yang lain, menjadikannya tak dapat dibaca oleh pencegat atau penyadap tanpa ilmu khusus (di mana sandi dibutuhkan untuk dekripsi pesan itu). Enkripsi digunakan untuk menyakinkan kerahasiaan di komunikasi, termasuk teknik untuk pemeriksaan integritas pesan, autentikasi identitas pengirim/penerima, tanda-tangan digital, bukti interaktif dan komputasi keamanan, serta banyak lagi yang lain.

- **Kriptografi Klasik**

  Bentuk awal dari penulisan rahasia membutuhkan lebih sedikit dari implementasi penulisan sejak banyak orang tidak dapat membaca. lawan yang lebih terpelajar, membutuhkan kriptografi yang nyata. Tipe sandi klasik utama ialah sandi transposisi, di mana mengatur aturan huruf pada pesan (contoh 'hello world' menjadi 'ehlol owrdl' pada skema pengubahan sederhana ini), dan sandi subtitusi, di mana secara sistematis mengganti huruf atau grup kata dengan kata lainnya dari grup kata (contoh 'fly at once' menjadi 'gmz bu podf' dengan mengganti setiap huruf dengan yang lain di alfabet Latin. Substitusi sandi pada awalnya disebut sandi Caesar, di mana setiap kata pada teks diganti dengan huruf dari jumlah tetap pada posisi di alfabet. Laporan Suetonius menyebutkan Julius Caesar mengunakannya untuk berkomunikasi dengan jendral-jendralnya. Atbash adalah contoh dari sandi Ibrani pada mulanya. Penggunaan awal kriptografi yang diketahui merupakan teks sandi yang diukir pada batu di Mesir (1900 sebelum Masehi), tetapi teks sandi ini digunakan hanya sebagai hiburan untuk pengamat terpelajar daripada cara untuk menyimpan informasi.

  Yunani kuno menyebutkan telah mengetahui sandi (contoh sandi transposisi scytale yang diklaim telah digunakan oleh militer Sparta. Steganografi (menyembunyikan kehadiran pesan sehingga pesan tersebut menjadi rahasia) juga pertama kali diperkenalkan pada masa kuno. Contoh awal seperti, dari Herodotus, menyembunyikan pesan - sebuah tato pada kepala budaknya - di bawah rambut yang kembali tumbuh. Contoh yang lebih modern dari steganografi termasuk penggunaan tinta tak tampak, mikrodot, dan tanda air digital untuk menyembunyikan informasi.

  Di India, Kamasutra dari Vātsyāyana yang berumur 2000 tahun berbicara dengan dua jenis sandi yang berbeda yang disebut Kautiliyam dan Mulavediya. Di Kautiliyam, substitusi kata sandi berdasarkan relasi fonetik, seperti vokal menjadi konsonan. Di Mulavediya, alfabet sandi terdiri dari kata-kata yang berpasangan dan bertimbal-balik.

Teks sandi yang dihasilkan dengan sandi klasik (dan beberapa sandi modern) selalu mengungkapkan informasi statistik tentang teks awal, yang sering dapat digunakan untuk memecahkannya. Setelah ditemukannya analisis frekuensi oleh matematikawan Arab dan polymath Al-Kindi (juga dikenal sebagai Alkindus) pada abad ke-9, hampir semua jenis sandi menjadi lebih sulit dipecahkan oleh penyerang yang memiliki informasi tersebut. Seperti sandi klasik yang masih populer hingga saat ini, meskipun lebih banyak dalam bentuk puzzle. Al-Kindi menuliskan buku kriptografi yang berjudul Risalah fi Istikhraj al-Mu'amma (Risalah untuk Menerjemahkan Pesan Kriptografi), yang menjelaskan teknik analisis frekuensi kriptanalisis yang pertama kalinya.

Pada dasarnya semua sandi tetap rentan kepada kriptanalisis menggunakan teknik analisis frekuensi hingga pengembangan dari sandi polyalphabetic, yang dijelaskan oleh Leon Battista Alberti sekitar tahun 1467, meskipun terdapat beberapa indikasi bahwa hal ini telah terlebih dahulu diketahui oleh Al-Kindi.[13] Penemuan Alberti menggunakan sandi yang berbeda (seperti subtitusi alfabet) untuk beberapa bagian pesan (mungkin untuk setiap teks surat berturut-turut hingga akhir). Dia juga menemukan apa yang mungkin menjadi alat sandi ototmatis untuk pertama kalinya, roda yang menerapkan pelaksanaan dari penemuannya. Pada sandi Vigenère polyalphabetic, enkripsi menggunakan kata kunci, yang mengatur substitusi surat berdasarkan surat mana dari kata kunci yang digunakan. Pada pertengahan abad ke-19 Charles Babbage menunjukkan bahwa sandi Vigenère sangat rentan terhadap pemeriksaan Kasiski, tetapi hal ini diterbitkan pertama sekali kira-kira sepuluh tahun kemudian oleh Friedrich Kasiski.

Walaupun analisis frekuensi dapat sangat kuat dan menjadi teknik umum melawan banyak sandi, enskripsi masih sangat efektif dalam penerapannya, sebagaimana banyak kriptanalisis masih khawatir akan penerapannya. Memecahkan pesan tanpa menggunakan analisis frekuensi pada dasarnya membutuhkan pengetahuan sandi dan mungkin kunci yang digunakan, sehingga membuat spionase, penyuapan, pencurian, dll. Hal ini secara tegas mengakui kerahasiaan algoritme sandi pada abad 19 sangat tidak peka dan tidak menerapkan praktik keamanan pesan; faktanya, hal ini lebih lanjut disadari bahwa setiap skema kriptografi yang memadai (termasuk sandi) harus tetap aman walaupun musuh benar-benar paham tentang algoritme sandi itu sendiri. Keamanan kunci yang digunakan harus dapat menjamin keamanan pemegang kunci agar tetap rahasia bahkan ketika diserang sekalipun. Prinsip fundamental ini pertama kali dijelaskan pada tahun 1883 oleh Auguste Kerckhoffs dan secara umum dikenal dengan Prinsip Kerckhoff; secara alternatif dan blak-blakan, hal ini dijelaskan kembali oleh Claude Shannon, penemu teori informasi dan fundamental dari teori kriptografi, seperti pribahasa Shanon - 'musuh mengetahui sistemnya'.

Alat-alat bantu yang berbeda telah banyak digunakan untuk membantu sandi. Salah satu alat paling tua yang dikenali merupakan scytale dari Yunani, tangkai yang digunakan oleh Spartan sebagai alat bantuk untuk memindahkan sandi (lihat gambar di atas). Pada zaman pertengahan, alat bantu lainnya ditemukan seperti jerejak sandi, yang juga dikenal sebagai jenis steganografi. Dengan penemuan polialfabetik, sandi menjadi lebih mutakhir dengan

bantuan disk sandi milik Alberti, skema tabula recta Johanner Trithemius, dan silinder multi Thomas Jefferson (tidak banyak diketahui, dan ditemukan kembali oleh Bazeries sekitar tahun 1900. Banyak alat mekanik enkripsi/dekripsi ditemukan pada awal abad ke-20, dan beberapa telah dipatenkan, di antaranya mesin rotor yang dikenal dengan nama mesin Enigma digunakan oleh pemerintah dan militer Jerman dari akhir tahun 1920-an dan selama Perang Dunia II. The ciphers implemented by better quality examples of these machine designs brought about a substantial increase in cryptanalytic difficulty after WWI.

- **Era Komputer**

  Kriptanalisis dari alat mekanis baru terbukti lebih sulit dan melelahkan. Di Inggris, usaha kriptanalisis di Bletchley Park selama Perang Dunia II memacu perkembangan alat yang lebih efisien untuk membawa tugas yang berulang-ulang. Hal ini berujung pada pengembangan Colossus, komputer digital pertama sekali yang bekerja penuh secara elektronik, yang membantu penyandi untuk mendekripsikan mesin Lorenz SZ40/42 milik tentara Jerman.

  Seperti juga pengembangan komputer digital dan elektronik, kriptanalisis juga berkembang menjadi lebih kompleks. Lebih jauh lagi, komputer dapat mengenkripsi setiap jenis data yang mungkin dalam bentuk biner, tidak seperti chiper klasik yang hanya mengenkripsi teks bahasa tertulis; hal ini sangat baru dan signifikan. Penggunaan komputer telah menggantikan kirptografi bahasa, baik untuk desain chiper dan kriptanalisis. Banyak chiper komputer dapat dikategorikan dengan operasi mereka pada urutan biner (kadang dalam grup atau blok), tidak seperti skema mekanikal dan klasik, yang biasa memanipulasikan karakter tradisional (seperti surat dan digit) secara langsung. Bagaimanapun, komputer juga membantu kriptanalisis, yang mengimbangi hingga batas tertentu untuk kompleksitas chiper yang lebih tinggi. Namun, chiper modern yang baik telah mengungguli kriptanalisis; kasus ini biasa menggunakan kualitas chiper yang lebih efisien (seperti membutuhkan sumber daya yang sedikit dan cepat, seperti memori atau kapabilitas CPU), sedang mendekripsikannya membutuhkan udaha dengan urutan yang lebih besar, dan lebih luas dibandingkan chiper yang lebih klasik, membuat kriptanalsis menjadi lebih tidak efisien dan tidak berguna.

  Riset akademik kriptografi yang terbuka luas masih relatif baru; dimulai pada pertengahan 1970-an. Pada saat ini, personel IBM mendesain algoritme yang menjadi standar enkripsi data Federal; Whitfield Diffie dan Martin Hellman mempublikasikan algoritme perjanjian mereka.[17] dan algoritme RSA yang dipublikasikan oleh kolumnis Amerika Martin Gardner. Sejak saat itu, kriptografi telah sangat luas digunakan dalam dunia komunikasi, jaringan komputer, dan keamanan komputer secara umum. Beberapa teknik kriptografi modern dapat menyimpan kunci rahasianya jika menggunakan masalah matematika rumit, seperti faktorisasi integer atau masalah logaritma diskrit, jadi terdapat hubungan yang mendalam dengan matematika abstrak. Tidak ada bukti pasti bahwa teknik kriptografi aman. Saat terbaik, terdapat bukti bahwa beberapa teknik aman jika beberapa masalah

komputasional sulit untuk dipecahkan, atau asumsi tertentu mengenai implementasi atau penggunakan praktikal bertemu.

Semakin mengetahui sejarah kriptografi, algoritme kriptografi dan desainer sistem harus juga bijaksana mempertimbangkan pengembangan masa depan saat bekerja dengan desain mereka. Sebagai contoh, pengembangan kontinu pada computer processing power telah meningkatkan cakupan brute-force attack, jadi ketika menentukan panjang kunci, diharuskan memilih kunci yang sulit.[18] Efek potensial dari komputer kuantum telah dipertimbangkan oleh beberapa sistem desainer kriptografi; implementasi kecil dari mesin ini yang akan segera diumumkan mungkin akan membutuhkan perhatian khusus ketimbang hanya spekulatif.[4]

Pada Dasarnya, pada awal abad ke-20, kriptografi secara singkat berkenaan dengan bahasa dan pola lexicographic. Sejak saat itu tekanannya telah berubah, dan kriptografi saat ini lebih luas menggunakan matematika, termasuk aspek teori informasi, kompleksitas komputasional, statistik, kombinasi, aljabar, teori bilangan, dan matematika diskrit secara umum. Kriptografi juga merupakan cabang teknik, tetapi tidak biasa sebab hal ini berkaitan dengan pertentangan yang aktif, pintar, dan jahat; teknik jenis lain (seperti teknik kimia dan sipil) hanya membutuhkan gaya natural netral. Terdapat juga riset berkembang yang memeriksa hubungan antara masalah kriptografi dan fisika kuantum.

1.3 Tujuan dan Manfaat Kriptografi

Adapun Tujuan dari kriptografi antara lain sebagai beriku.

1. Kerahasiaan

Salah satu tujuan utama mengapa seseorang membuat pesan dengan berbagai enkripsi adalah kerahasiaan. Pesan dibuat dengan sedemikian rupa agar tidak mudah dibaca oleh pihak ketiga yang dianggap tidak perlu atau menjadi pihak yang memang harus dihindari.

Ketika membuat pesan yang memuat rahasia, akan ada kode atau kunci yang digunakan sebagai petunjuknya. Misal dengan menyimbolkan angka dengan huruf atau sejenisnya. Selain itu, juga bisa menggunakan aneka simbol abstrak lain yang sesuai dengan kebutuhan.

Intinya, kerahasiaan akan menjadi pokok utama yang harus diperhatikan dengan baik. Bahkan, ini juga bisa digunakan untuk membuat lapisan enkripsi yang lebih dari satu. Jadi, seseorang harus beberapa kali membuka kunci agar mengetahui pesan atau isi asli yang disamarkan.

2. Autentikasi

Saat ini autentikasi mulai banyak digunakan di berbagai akun media sosial atau akun perbankan. Ini dipakai untuk melakukan pengenalan terhadap orang yang memiliki atau diberikan akses untuk membuka dan melakukan modifikasi pada akun yang sedang digunakan.

Proses enkripsi dengan cara ini biasanya dilakukan dengan kombinasi username dan kata kunci. Selain itu juga bisa menggunakan nomor telepon, OTP, atau lainnya. Intinya autentikasi akan ditujukan untuk meyakinkan sistem jika orang yang mengakses benar-benar asli.

Adanya autentikasi yang dilakukan, akan sulit bagi pihak ketiga untuk masuk ke dalamnya. Apalagi autentikasi akan dilakukan dua arah. Kemungkinan besar akan menyulitkan orang lain untuk bisa masuk ke dalam akun dan membaca seluruh isi di dalamnya.

3. Integritas Data

Integritas dari data yang nantinya akan diterima juga merupakan bagian terpenting dari kriptografi ini sendiri. Karena kita tidak akan tahu kira-kira data yang dikirim orang lain itu benar-benar dari dia atau ada orang lain yang menyamar dan mengirim data palsu.

Biasanya akan ada platform tertentu yang digunakan untuk pengiriman data. Hanya dua orang atau lebih yang mengetahuinya. Apabila data akhirnya dimasukkan ke sana, kemungkinan besar akan mudah untuk diketahui integritasnya.Ini akan menghindari adanya kecurigaan yang tidak penting. Jadi, antara orang satu dengan yang lain akan saling percaya. Jadi, proses pengiriman dan penerimaan data bisa berjalan dengan baik.

4. Pertukaran Kunci

Disadari atau tidak, sebenarnya dalam proses enkripsi atau penerapan kriptografi ini ada metode pertukaran kunci. Seseorang akan mengirimkan data lalu orang lain yang menerima akan membukanya sesuai kunci yang diberikan. Apabila terjadi komunikasi dua arah, kemungkinan besar akan saling menukar kunci secara berkala. Pesan yang terenkripsi hanya bisa digunakan untuk mereka yang saling terhubung dan mendapatkan kunci saja.

Sementara itu, orang lain yang tidak ada kaitannya sama sekali akan sulit untuk membukanya. Biasanya proses ini hanya bisa dilakukan kalau dua orang terhubung dalam platform yang sama. Misal pada pesan singkat atau chat yang hanya dua orang ini saja yang mengetahui isinya.

5. Non Repudiasi

Saat melakukan sesuatu pada suatu objek di internet, biasanya akan ada jejak siapa saja yang sudah masuk ke sana, mengganti sesuatu, atau malah melakukan penghapusan. Bukti ini akan tersimpan dan bisa dicek ulang untuk melakukan pembuktian. Dalam kriptografi, hal ini sangat penting dan umumnya bersifat rahasia. Perekaman dilakukan dengan tujuan untuk menghindari adanya prasangka. Jadi, kalau di kemudian hari ada masalah yang terjadi bisa segera diatasi tanpa menyebabkan konflik yang besar.

Beberapa aplikasi sudah mendukung ini sehingga bisa melakukan tracking aktivitas yang sudah dilakukan oleh seseorang. Misal pada aplikasi Docs di Google Drive. Siapa pun yang melakukan penyuntingan file akan tercatat termasuk dengan perubahannya. Itulah beberapa tujuan dari kriptografi yang perlu dipahami. Terdapat lima tujuan yang memiliki penjelasan masing-masing.

BAB II
PRINSIP DASAR ENKRIPSI

2.1 Enkripsi Simetris

Enkripsi simetris adalah salah satu teknik kriptografi yang paling umum digunakan di mana kunci yang sama digunakan baik untuk mengenkripsi maupun mendekripsi data. Ini berbeda dengan enkripsi asimetris di mana terdapat dua kunci terpisah untuk enkripsi dan dekripsi.

Cara kerja enkripsi simetris dapat dijelaskan sebagai berikut:

- Pemilihan Algoritma: Pertama, pengguna memilih algoritma enkripsi yang akan digunakan, seperti AES, DES, atau 3DES. Setiap algoritma memiliki karakteristik dan kekuatan masing-masing.
- Pemilihan Kunci: Pengguna kemudian memilih kunci yang akan digunakan untuk enkripsi dan dekripsi. Kunci ini bisa berupa urutan bit atau karakter yang diperlukan oleh algoritma enkripsi tertentu.
- Enkripsi: Dalam proses enkripsi, teks biasa atau plaintext diubah menjadi teks terenkripsi atau ciphertext menggunakan algoritma enkripsi yang dipilih dan kunci yang telah ditentukan. Proses ini melibatkan serangkaian operasi matematis tergantung pada algoritma yang digunakan.
- Dekripsi: Ketika pesan terenkripsi harus dibaca, proses dekripsi dilakukan. Ini melibatkan penggunaan kunci yang sama yang digunakan dalam enkripsi untuk mengembalikan ciphertext menjadi plaintext.

Karakteristik utama dari enkripsi simetris adalah sebagai berikut:

- Efisiensi Komputasi: Enkripsi simetris cenderung lebih efisien secara komputasi dibandingkan dengan enkripsi asimetris. Operasi yang digunakan relatif sederhana dan memungkinkan untuk enkripsi dan dekripsi data dalam jumlah besar dengan cepat.
- Kecepatan: Karena efisiensi komputasinya, enkripsi simetris dapat mengenkripsi dan mendekripsi data dalam jumlah besar dengan cepat, membuatnya ideal untuk aplikasi yang membutuhkan pemrosesan data yang cepat.
- Pertukaran Kunci: Salah satu kelemahan utama enkripsi simetris adalah perlunya pertukaran kunci yang aman antara pihak yang berkomunikasi. Kunci yang sama harus

disepakati oleh kedua belah pihak sebelum komunikasi dimulai. Ini memerlukan metode yang aman untuk pertukaran kunci, karena jika kunci jatuh ke tangan yang salah, kerahasiaan data dapat terancam.

Contoh algoritma enkripsi simetris yang umum digunakan termasuk:

- Advanced Encryption Standard (AES): AES adalah salah satu algoritma enkripsi simetris yang paling umum digunakan. Ini memiliki kunci dengan panjang 128, 192, atau 256 bit dan telah diadopsi secara luas di berbagai aplikasi.
- Data Encryption Standard (DES): DES adalah salah satu algoritma enkripsi simetris yang lebih tua. Meskipun sekarang dianggap tidak aman karena panjang kuncinya yang terlalu pendek, namun masih digunakan dalam beberapa konteks tertentu.
- Triple DES (3DES): 3DES adalah variasi dari DES yang lebih aman. Ini menggunakan kunci dengan panjang 168 bit dan menerapkan DES tiga kali untuk meningkatkan keamanan.

Algoritma-algoritma ini telah diuji dan digunakan secara luas dalam aplikasi-aplikasi keamanan informasi, mulai dari pengamanan pesan-pesan pribadi hingga transaksi keuangan online yang sensitif.

2.2 Enkripsi Asimetris

Enkripsi asimetris, atau kriptografi kunci publik, adalah teknik kriptografi yang menggunakan sepasang kunci yang berbeda untuk enkripsi dan dekripsi pesan. Kunci publik digunakan untuk mengenkripsi pesan, sementara kunci privat digunakan untuk mendekripsinya. Teknik ini dikenal sebagai "asimetris" karena kunci publik dan kunci privat tidak identik, berbeda dengan enkripsi simetris di mana kunci yang sama digunakan untuk kedua proses.

Cara Kerja:

- Enkripsi: Pengirim menggunakan kunci publik penerima (yang biasanya tersedia secara publik) untuk mengenkripsi pesan sebelum mengirimkannya. Ini memastikan bahwa pesan hanya dapat dibaca oleh penerima yang memiliki kunci privat yang sesuai.
- Dekripsi: Penerima menggunakan kunci privatnya (yang hanya dia yang memiliki) untuk mendekripsi pesan yang telah dienkripsi. Hanya penerima yang memiliki kunci

privat yang sesuai dengan kunci publik yang digunakan untuk enkripsi yang dapat membaca pesan.

Karakteristik:

- Kompleksitas Algoritma: Enkripsi asimetris biasanya membutuhkan waktu dan sumber daya komputasi yang lebih besar daripada enkripsi simetris karena algoritma yang digunakan lebih kompleks.
- Pertukaran Kunci yang Aman: Enkripsi asimetris tidak memerlukan pertukaran kunci yang aman sebelumnya karena kunci publik dapat dibagikan secara terbuka. Ini membuatnya cocok untuk penggunaan di lingkungan di mana pertukaran kunci yang aman tidak mungkin, seperti internet.
- Keamanan: Enkripsi asimetris memberikan tingkat keamanan yang tinggi karena hanya penerima yang memiliki kunci privat yang dapat mendekripsi pesan. Selain itu, proses tanda tangan digital juga menggunakan enkripsi asimetris untuk memverifikasi keaslian pesan.

Contoh Algoritma:

- RSA (Rivest-Shamir-Adleman): RSA adalah salah satu algoritma enkripsi asimetris yang paling umum digunakan. Ini berdasarkan kompleksitas matematika dalam hal faktorisasi bilangan bulat besar.
- Elliptic Curve Cryptography (ECC): ECC adalah teknik kriptografi yang menggunakan kurva elips untuk menghasilkan pasangan kunci. Ini menawarkan tingkat keamanan yang tinggi dengan menggunakan kunci yang lebih pendek dibandingkan dengan RSA.
- Digital Signature Algorithm (DSA): DSA adalah algoritma yang digunakan untuk menghasilkan tanda tangan digital yang sah. Ini sering digunakan dalam aplikasi yang memerlukan otentikasi pesan, seperti transaksi keuangan online.

2.3 Proses Enkripsi

Proses enkripsi dan dekripsi dapat dijelaskan secara matematis sebagai berikut:

1. Proses Enkripsi :
    - Plaintext (M): Pesan asli yang ingin dikirim atau disimpan.

- Fungsi Enkripsi (E): Fungsi atau algoritma yang digunakan untuk mengubah plaintext menjadi ciphertext.
- Ciphertext (C): Hasil dari proses enkripsi, yaitu pesan yang sudah teracak dan tidak dapat dibaca tanpa proses dekripsi.

Secara matematis, proses enkripsi dapat dituliskan sebagai:

$$E(M) = C$$

Di mana:

- M adalah plaintext (pesan asli).
- C adalah ciphertext (pesan terenkripsi).

2. Proses Dekripsi:
   - Ciphertext (C): Pesan terenkripsi yang diterima atau disimpan.
   - Fungsi Dekripsi (D): Fungsi atau algoritma yang digunakan untuk mengembalikan ciphertext ke bentuk plaintext.
   - Plaintext (M): Pesan asli yang telah berhasil didekripsi.

Secara matematis, proses dekripsi dapat dituliskan sebagai:

$$D(C) = M$$

Di mana:

- C adalah ciphertext (pesan terenkripsi).
- M adalah plaintext (pesan asli).

Untuk menggambarkan alur lengkap dari proses enkripsi dan dekripsi:

$$\text{Plaintext}(M) \underset{D}{\overset{E}{\rightleftarrows}} \text{Ciphertext}(C)$$

Dengan kata lain, prosesnya adalah:

1. **Enkripsi** : Plaintext M dienkripsi menggunakan fungsi enkripsi E untuk menghasilkan ciphertext C.
2. **Dekripsi** : Ciphertext C didekripsi menggunakan fungsi dekripsi D untuk mengembalikan plaintext M.

Contoh Kasus :

Misalkan kita menggunakan algoritma enkripsi tertentu, katakanlah algoritma substitusi sederhana, di mana setiap huruf dalam plaintext digantikan dengan huruf lain berdasarkan kunci tertentu.

- **Plaintext (M)** : "HELLO"
- **Fungsi Enkripsi (E)** : Misalkan substitusi menggunakan kunci sederhana di mana H→X, E→Y, L→Z, O→W.
- **Ciphertext (C)** : "XYZZW"

Jika kita mengirim "XYZZW" sebagai ciphertext, penerima yang memiliki kunci enkripsi dapat menggunakan fungsi dekripsi untuk mengembalikannya ke plaintext asli "HELLO".

2.4 Teknik Dasar Kriptografi

Berikut merupakan beberapa teknik dasar dari proses enkripsi yaitu.

2.4.1 Substitusi

Teknik substitusi adalah salah satu teknik dasar dalam kriptografi yang melibatkan penggantian elemen-elemen dari plaintext dengan elemen-elemen lain untuk menghasilkan ciphertext. Ada beberapa bentuk teknik substitusi, tetapi yang paling umum adalah substitusi sederhana di mana setiap huruf dalam plaintext digantikan dengan huruf lain berdasarkan suatu aturan atau kunci tertentu.

Pada teknik dasar kriptografi substitusi, **pesan asli (*plainteks*)** diubah menjadi **pesan terenkripsi (*chiperteks*)** dengan cara **mengganti setiap huruf, simbol, atau kelompok karakter dalam plainteks dengan karakter lain yang telah ditentukan**. Teknik ini bertujuan untuk **menyembunyikan isi pesan** dari pihak yang tidak berwenang.

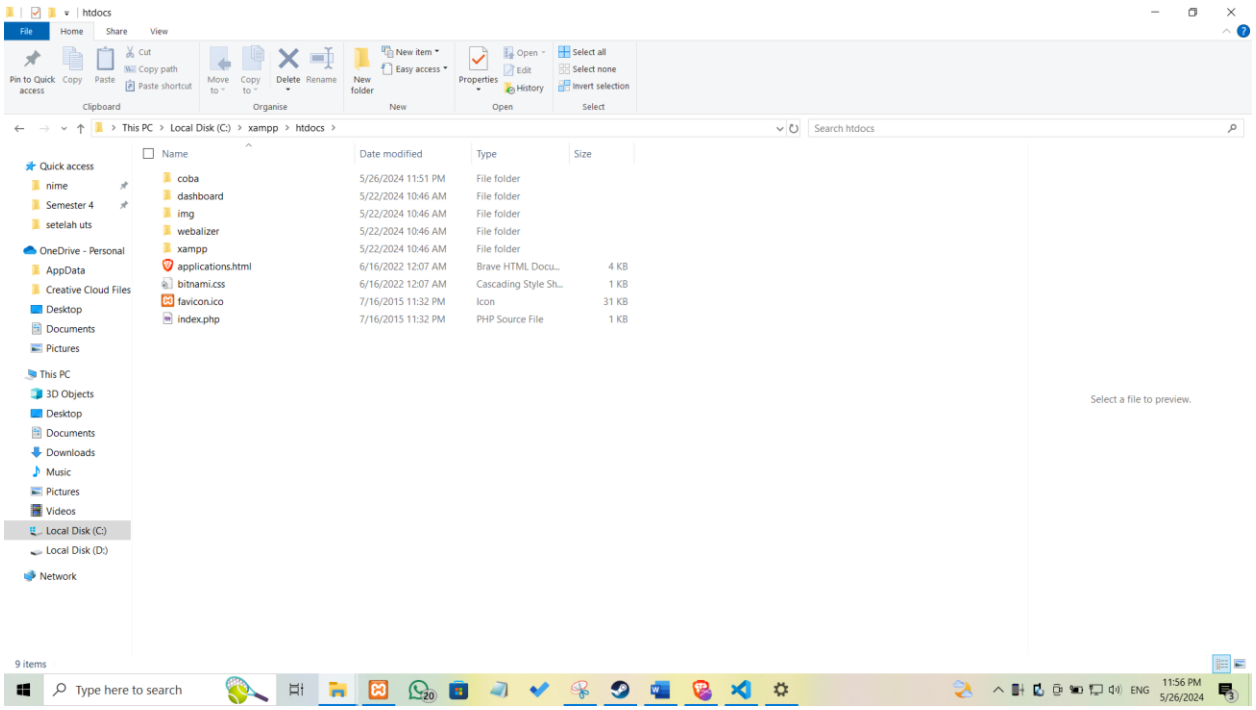
Langkah-langkah *Substitusi*

1. Langkah pertama adalah membuat suatu tabel substitusi. Tabel substitusi dapat dibuat sesuka hati, dengan catatan bahwa penerima pesan memiliki tabel yang sama untuk keperluan dekripsi.
2. Bila tabel substitusi dibuat secara acak, akan semakin sulit pemecahan ciphertext oleh orang yang tidak berhak, contoh.
   - Tabel Substitusi

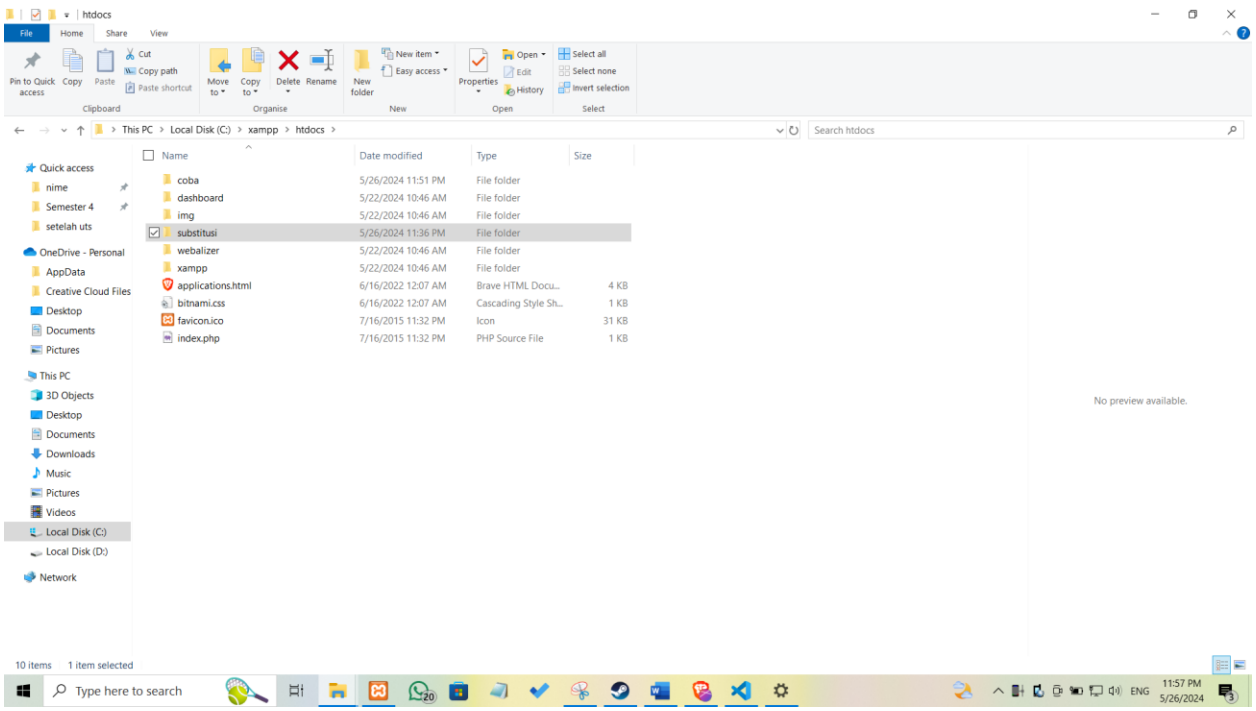
- Caesar Chipher
- ROT 13

Langkah-langkah praktikum:

1. Buatlah folder di penyimpanan xampp/htdocs (default ada di C:\xampp\htdocs).

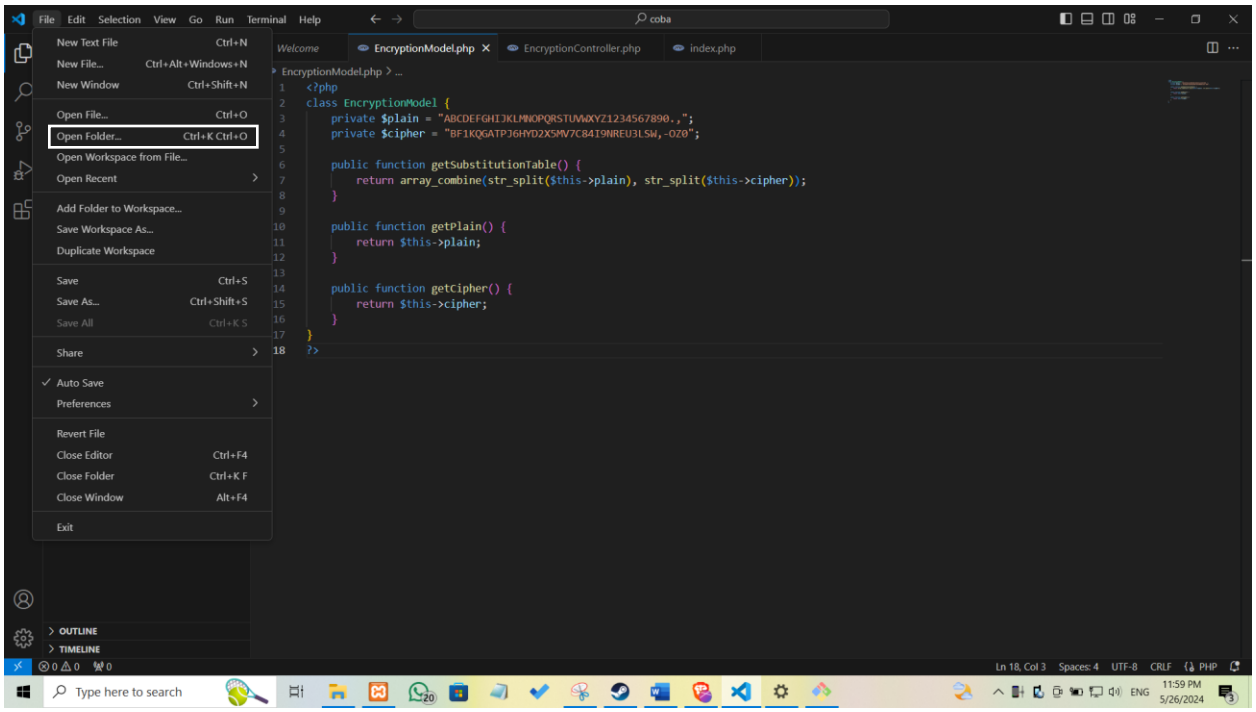


2. Selanjutnya buatlah folder dengan nama Substitusi

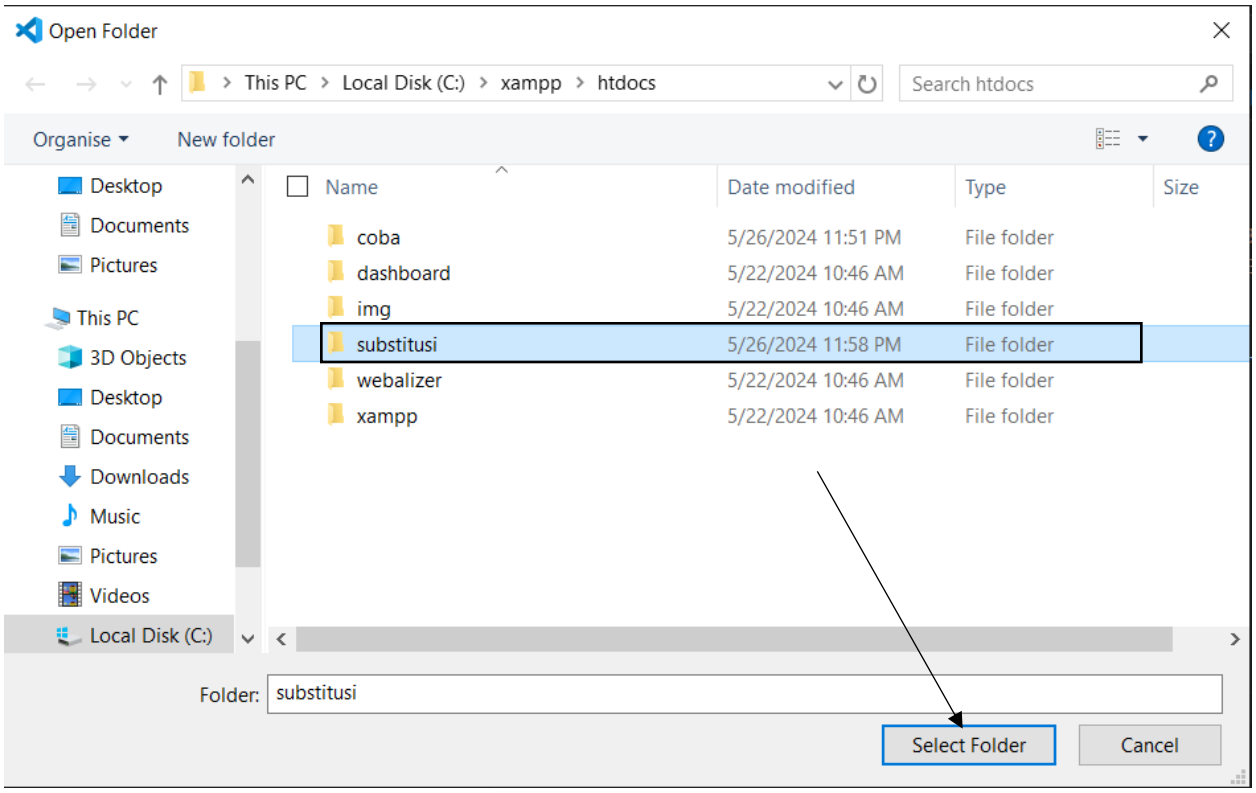


3. Buka folder substitusi di visual studio code(jika belum bisa silahkan ikuti Langkah berikut)

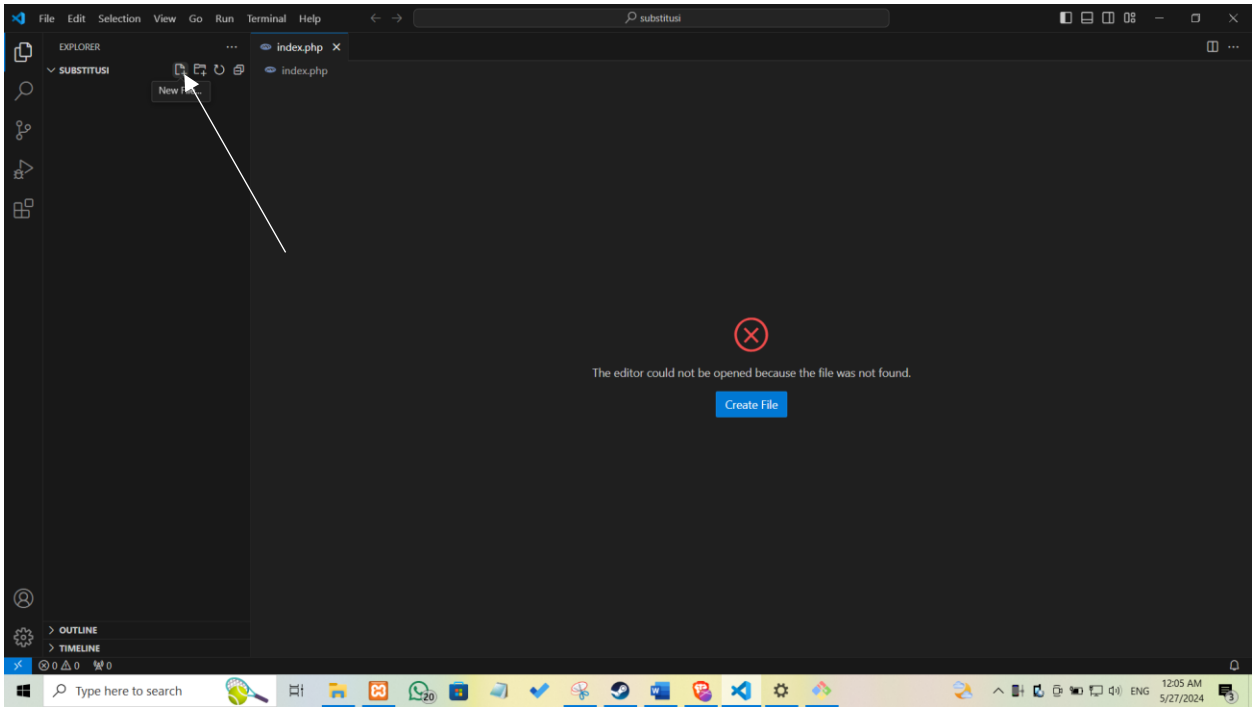
Klik file dan open folder atau shortcut (ctrl+k+ctrl+o)



Buka folder ini C:\xampp\htdocs\substitusi di vscode



4. Buat file dengan cara klik ini

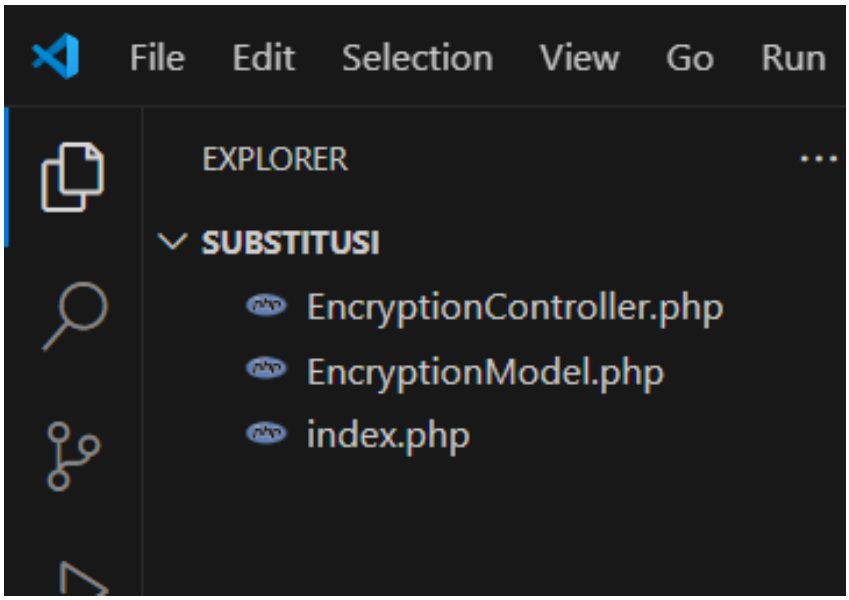


5. Buat 3 file berikut

EncryptionController.php

EncryptionModel.php

index.php



6. Dalam EncryptionController.php masukkan code berikut

```
<?php
class EncryptionController {
    private $model;

    public function __construct($model) {
        $this->model = $model;
    }

    public function substitusi($input) {
        $substitution = $this->model->getSubstitutionTable();
        $output = "";
        for ($i = 0; $i < strlen($input); $i++) {
$char = strtoupper($input[$i]);
            if (isset($substitution[$char])) {
```

```
                $output .= $substitution[$char];
            } else {
                $output .= $char;
            }
        }
        return $output;
    }

    public function caesarCipher($input, $shift = 3) {
        $output = '';
        $input = strtoupper($input);
        $alphabet = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ';
        for ($i = 0; $i < strlen($input); $i++) {
            $char = $input[$i];
            $pos = strpos($alphabet, $char);
            if ($pos !== false) {
                $newPos = ($pos + $shift) % 26;
                $output .= $alphabet[$newPos];
            } else {
                $output .= $char;
            }
        }
        return $output;
    }

    public function rot13($input) {
        return str_rot13($input);
    }
}
?>
```

7. Dalam EncryptionModel.php masukkan code berikut

```
<?php
class EncryptionModel {
    private $plain = "ABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890.,";
    private $cipher = "BF1KQGATPJ6HYD2X5MV7C84I9NREU3LSW,-OZ0";

    public function getSubstitutionTable() {
        return array_combine(str_split($this->plain), str_split($this->cipher));
    }

    public function getPlain() {
        return $this->plain;
    }

    public function getCipher() {
        return $this->cipher;
    }
}
?>
```

8. Dalam index.php masukkan code berikut

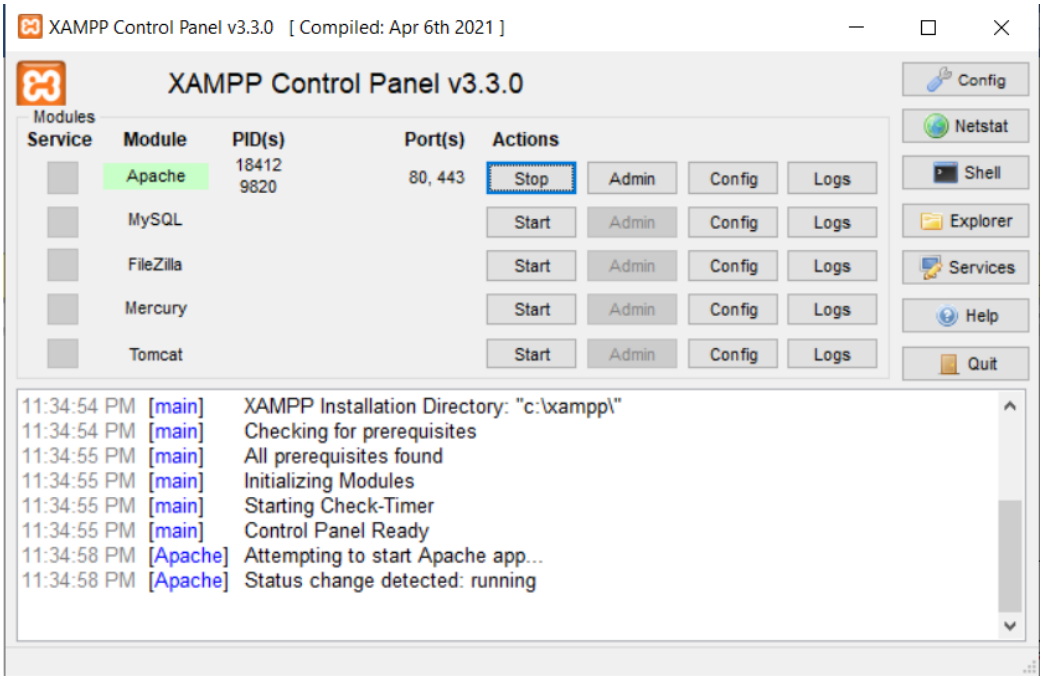
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Enkripsi</title>
    <style>
        body {
```

```
        font-family: Arial, sans-serif;
        background-color: #f0f0f0;
        margin: 20px;
    }
    .container {
        max-width: 600px;
        background-color: #fff;
        padding: 20px;
        border-radius: 5px;
        box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    }
    .container h2 {
        color: #333;
    }
    form {
        margin-bottom: 20px;
    }
    label {
        font-weight: bold;
    }
    input[type="text"] {
        width: 100%;
        padding: 10px;
        margin: 5px 0 10px;
        border: 1px solid #ccc;
        border-radius: 4px;
        box-sizing: border-box;
    }
    input[type="submit"] {
        background-color: #4CAF50;
        color: white;
```

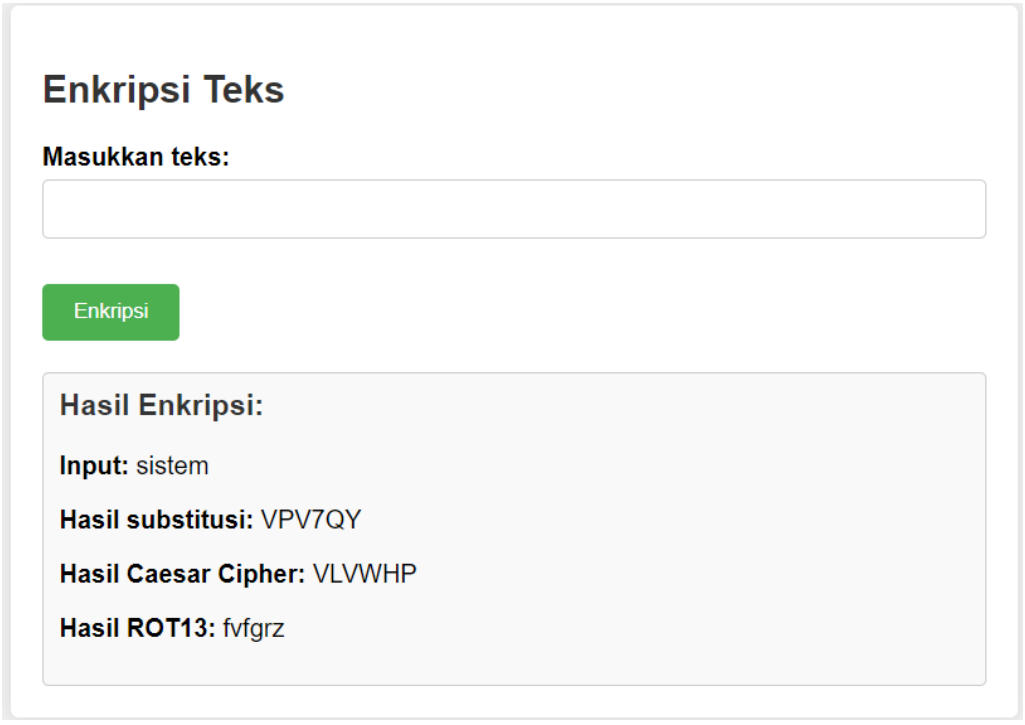
```
            padding: 10px 20px;
            border: none;
            border-radius: 4px;
            cursor: pointer;
        }
        input[type="submit"]:hover {
            background-color: #45a049;
        }
        .result {
            background-color: #f9f9f9;
            padding: 10px;
            border: 1px solid #ccc;
            border-radius: 4px;
            margin-top: 10px;
        }
        .result h3 {
            color: #333;
            margin-top: 0;
        }
    </style>
</head>
<body>
    <div class="container">
        <h2>Enkripsi Teks</h2>
        <form method="post">
            <label for="input">Masukkan teks:</label>
            <input type="text" id="input" name="input" required>
            <br><br>
            <input type="submit" value="Enkripsi">
        </form>
```

```
        <?php
        require 'EncryptionModel.php';
        require 'EncryptionController.php';

        $model = new EncryptionModel();
        $controller = new EncryptionController($model);

        if ($_SERVER["REQUEST_METHOD"] == "POST") {
            $input = $_POST['input'];
            $substitusi = $controller->substitusi($input);
            $caesarCipher = $controller->caesarCipher($input);
            $rot13 = $controller->rot13($input);
        ?>
        <div class="result">
            <h3>Hasil Enkripsi:</h3>
            <p><strong>Input:</strong> <?php echo htmlspecialchars($input); ?></p>
            <p><strong>Hasil substitusi:</strong> <?php echo htmlspecialchars($substitusi);
?></p>
            <p><strong>Hasil Caesar Cipher:</strong> <?php echo
htmlspecialchars($caesarCipher); ?></p>
            <p><strong>Hasil ROT13:</strong> <?php echo htmlspecialchars($rot13); ?></p>
        </div>
        <?php } ?>
    </div>
</body>
</html>
```

9. Selanjutnya untuk menjalankan project, buka xampp dan pastikan apache nya menyala



10. Kemudian buka browser, dan ketikkan http://localhost/substitusi
11. Inilah hasil dari project Ketika dijalankan



Selain Cara tersebut kita dapat dengan menggunakan tabel substitusi

1. Langkah pertama adalah membuat suatu tabel substitusi. Tabel substitusi dapat dibuat sesuka hati, dengan catatan bahwa penerima pesan memiliki tabel yang sama untuk keperluan dekripsi.
2. Bila tabel substitusi dibuat secara acak, akan semakin sulit pemecahan ciphertext oleh orang yang tidak berhak.

   Contoh

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6	7	8	9	0	.

Dibawah adalah hasil dari enkripsi nya

B	F	1	K	Q	G	A	T	P	J	6	H	Y	D	2	X	5	M	V	7	C	8	4	I	9	N	R	E	U	3	L	S	W	.	O	Z	0

Misalnya kita memasukkan SISTEM

Maka hasilnya adalah 7P7CQY

- **Caesar Ciper**

Metode Caesar Cipher adalah salah satu teknik enkripsi paling sederhana dan terkenal dalam sejarah. Namanya diambil karena pemimpin militer dan politik Romawi Julius Caesar menggunakannya untuk mengirim pesan rahasia kepada para jenderalnya. Teknik ini merupakan jenis enkripsi substitusi di mana setiap huruf dari teks asli diganti dengan huruf lain yang beberapa tempat lebih rendah dalam alfabet.

Pada intinya, Caesar Cipher menggeser setiap huruf dalam pesan (plaintext) dengan jumlah tertentu dalam urutan alfabet. Julius Caesar konon menggunakan pergeseran sebanyak tiga posisi. Artinya, setiap huruf digantikan oleh huruf yang terletak tiga posisi setelahnya dalam alfabet. Contoh Enkripsi dari Caesar Cipher adalah sebagai berikut:

Plaintext (Teks Asli)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Ciphertext (Teks Hasil Enkripsi)

D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Contoh Enkripsi: Mari kita ambil contoh untuk mengilustrasikan bagaimana transformasi ini bekerja.

1. **Pesan Asli (Plaintext):** hello
2. **Proses Enkripsi:**
   - h digantikan oleh K (karena h adalah huruf ke-8, dan huruf ke-11 adalah K)
   - e digantikan oleh H (karena e adalah huruf ke-5, dan huruf ke-8 adalah H)
   - l digantikan oleh O (karena l adalah huruf ke-12, dan huruf ke-15 adalah O)
   - l digantikan oleh O (sama seperti sebelumnya)
   - o digantikan oleh R (karena o adalah huruf ke-15, dan huruf ke-18 adalah R)
3. **Pesan Terenkripsi (Ciphertext):** KHOOR

2.4.2 Blocking

Sistem enkripsi sering kali membagi plaintext (teks asli) menjadi blok-blok yang terdiri dari beberapa karakter. Blok-blok ini kemudian dienkripsi secara independen satu sama lain. Salah satu metode enkripsi yang umum digunakan adalah enkripsi blocking.

Pada enkripsi blocking, jumlah lajur (kolom vertikal) dan kolom (baris horizontal) untuk penulisan pesan ditentukan terlebih dahulu. Jumlah lajur atau kolom ini berfungsi sebagai kunci dalam teknik kriptografi ini. Berikut adalah langkah-langkah yang terlibat dalam proses enkripsi blocking:

1. Pembentukan Blok:
   – Plaintext dipecah menjadi blok-blok dengan ukuran yang sama sesuai dengan jumlah lajur yang telah ditentukan.

2. Penulisan Pesan:
   - Teks asli dituliskan secara vertikal dari atas ke bawah dalam setiap lajur. Setelah satu lajur penuh, penulisan dilanjutkan ke lajur berikutnya, dan proses ini berlanjut sampai seluruh teks asli tertulis.

3. Pembacaan Ciphertext:

- Setelah seluruh plaintext ditulis dalam lajur-lajur yang ditentukan, ciphertext (teks yang sudah dienkripsi) dibaca secara horizontal, dari kiri ke kanan dan dari atas ke bawah. Pembacaan ini mengikuti urutan blok yang telah dibentuk sebelumnya.

Jika plaintext adalah “5 TEKNIK DASAR KRIPTOGRAFI” maka hasil chipertext) . Jika menggunakan teknik blocking dengan 1 blok berisi 4 karakter.”

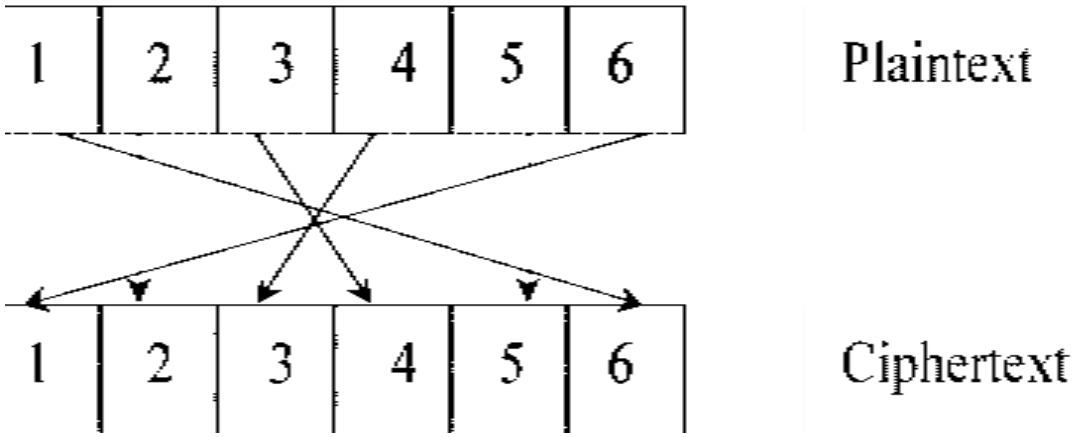
5	K		G	**BLOK 1**
		K	R	**BLOK 2**
T	D	R	A	**BLOK 3**
E	A	I	F	**BLOK 4**
K	S	P	I	**BLOK 5**
N	A	T		**BLOK 6**
I	R	O		**BLOK 7**

Jadi ciphertext yang dihasilkan dengan teknik ini adalah
"5K G KRTDRAEAIFKSPINAT IRO".
Plaintext dapat pula ditulis secara horizontal dan ciphertextnya adalah hasil pembacaan secara vertikal.

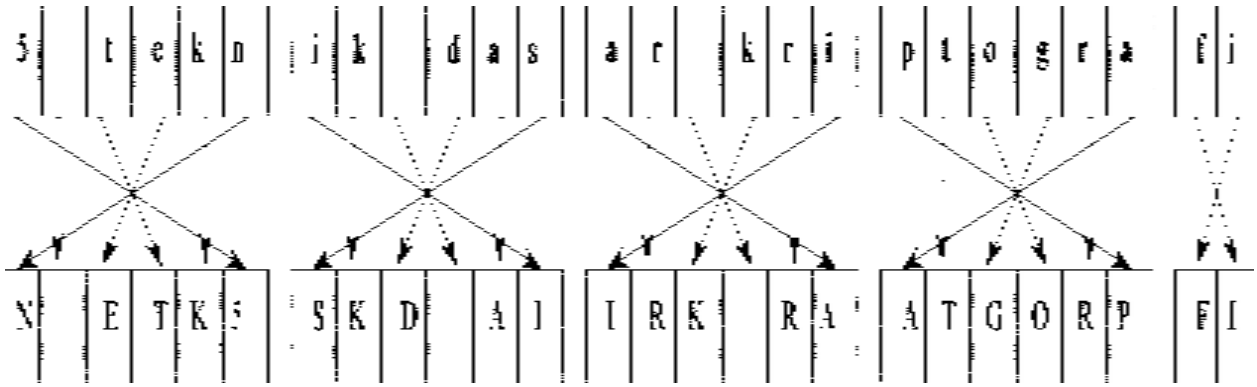
2.4.3 Permutasi

Salah satu teknik enkripsi yang penting dalam kriptografi adalah permutasi, yang juga dikenal sebagai transposisi. Teknik ini bekerja dengan cara mengubah urutan karakter-karakter dalam teks berdasarkan aturan tertentu, tanpa mengubah identitas karakter itu sendiri. Ini berarti bahwa karakter tetap sama, tetapi posisinya di dalam teks yang diubah. Teknik ini berbeda dengan teknik substitusi, di mana karakter-karakter diubah menjadi karakter lain tetapi posisinya tetap. Sebelum melakukan permutasi, plaintext atau teks asli biasanya dibagi terlebih dahulu menjadi blok-blok dengan panjang yang sama. Proses ini memastikan bahwa setiap bagian dari teks diacak posisinya secara sistematis, membuat pesan asli sulit dikenali tanpa kunci yang benar. Dengan cara ini, permutasi meningkatkan keamanan pesan karena meskipun seseorang mengetahui karakter-karakter yang digunakan, tanpa mengetahui aturan permutasinya, mereka tidak dapat

mengembalikan pesan ke bentuk aslinya. Teknik ini sering digunakan dalam berbagai algoritma enkripsi untuk meningkatkan kompleksitas dan keamanan data yang disampaikan.
Untuk contoh diatas, plaintext akan dibagi menjadi blok-blok yang terdiri dari 6 karakter, dengan aturan permutasi sebagai berikut :



Dengan menggunakan aturan diatas, maka proses enkripsi dengan permutasi dari plaintext adalah sebagai berikut :



Ciphertext yang dihasilkan dengan teknik permutasi ini adalah "N ETK5 SKD AIIRK RAATGORP FI".

2.4.4 Ekspansi

Suatu metode sederhana untuk mengacak pesan adalah dengan memelarkan pesan itu menggunakan aturan tertentu. Salah satu contoh penggunaan teknik ini adalah dengan meletakkan huruf konsonan atau bilangan ganjil yang menjadi awal dari suatu kata di akhir kata itu, kemudian menambahkan akhiran "an". Sebaliknya, bila suatu kata dimulai dengan huruf vokal atau bilangan genap, ditambahkan akhiran "i". Metode ini mengubah struktur asli pesan sehingga sulit dikenali tanpa mengetahui aturan pengacakannya, namun tetap menjaga elemen-elemen asli dari pesan sehingga bisa diuraikan kembali dengan aturan yang benar. Contohnya,

kata "pesan" menjadi "esanpan", dan kata "apel" menjadi "apeli". Dengan cara ini, pesan dapat disamarkan secara efektif, memberikan lapisan tambahan keamanan pada komunikasi.

Proses enkripsi dengan cara ekspansi terhadap plaintext terjadi sebagai berikut :

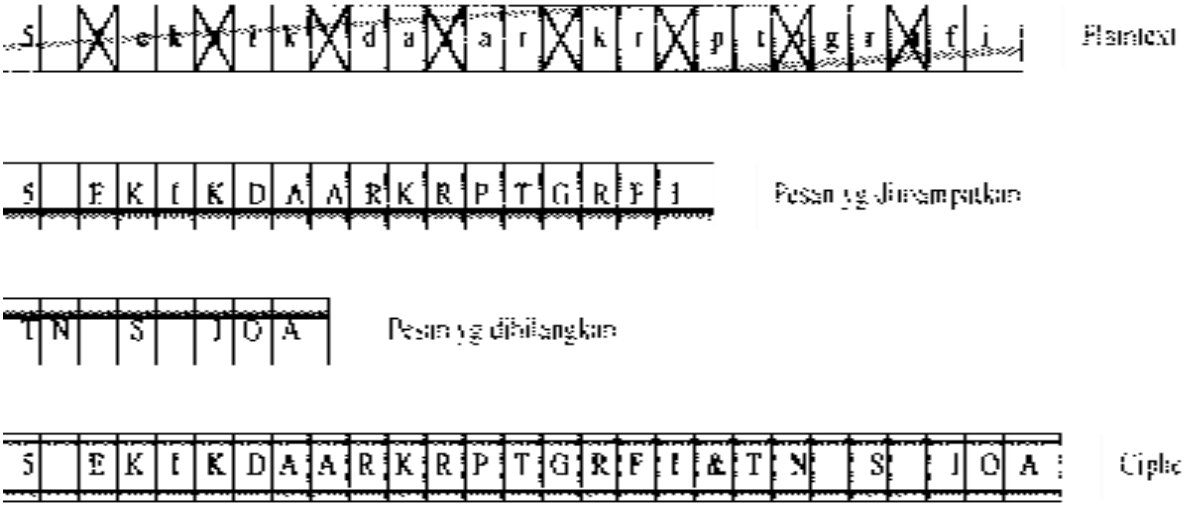
Ciphertextnya adalah

"5AN EKNIKTAN ASARDAN RIPTOGRAFIKAN".

2.4.5 Pemampatan

Mengurangi panjang pesan atau jumlah bloknya adalah cara lain untuk menyembunyikan isi pesan. Contoh sederhananya adalah dengan menghilangkan setiap karakter ke-tiga secara berurutan. Karakter-karakter yang dihilangkan kemudian disatukan kembali dan ditambahkan sebagai lampiran di akhir pesan utama, dengan diawali oleh karakter khusus, misalnya "&". Metode ini menyamarkan pesan asli dan menambahkan lapisan keamanan, membuatnya sulit dibaca tanpa mengetahui aturan pengurangannya.

Proses yang terjadi untuk plaintext kita adalah :



BAB III
ALGORITMA KRIPTOGRAFI POPULER

3.1 Advanced Encryption Standard (AES)

AES (Advanced Encryption Standard) adalah sebuah standar enkripsi yang digunakan untuk mengamankan data melalui proses penyandian (enkripsi) dan pendekripsiannya. Ini adalah metode enkripsi kunci simetris, yang berarti kunci yang digunakan untuk menyandikan pesan sama dengan kunci yang digunakan untuk mendekripsinya. AES digunakan secara luas di seluruh dunia untuk melindungi data sensitif seperti informasi pribadi, transaksi keuangan, dan komunikasi online.

Standar AES terdiri dari tiga varian utama: AES-128, AES-192, dan AES-256, yang masing-masing merujuk pada panjang kunci yang digunakan dalam proses enkripsi. Angka di belakang "AES" menunjukkan panjang kunci dalam bit, yaitu jumlah bit dalam kunci yang digunakan untuk menyandikan dan mendekripsi pesan. Varian AES-128 menggunakan kunci 128 bit, AES-192 menggunakan kunci 192 bit, dan AES-256 menggunakan kunci 256 bit.

Proses enkripsi AES melibatkan pembagian data menjadi blok-blok yang memiliki panjang tetap, yaitu 128 bit. Setiap blok kemudian disandikan menggunakan kunci yang telah ditentukan sebelumnya. Proses ini melibatkan serangkaian operasi matematis yang kompleks yang memanipulasi bit-bit data untuk menghasilkan data yang tidak terbaca. Proses pendekripsi mirip dengan proses enkripsi, tetapi menggunakan kunci yang sama untuk mengembalikan pesan asli dari data yang telah disandikan.

AES dianggap aman karena telah melalui analisis yang ketat oleh kriptografer dan ahli keamanan. Meskipun demikian, ketahanan AES terhadap serangan terus dimonitor dan dievaluasi secara berkala untuk memastikan keamanan sistem. Salah satu alasan AES dianggap aman adalah karena ukuran kunci yang besar, terutama pada varian AES-192 dan AES-256, yang membuatnya sulit untuk dipecahkan oleh serangan brute force, di mana penyerang mencoba setiap kemungkinan kombinasi kunci secara berurutan.

AES digunakan secara luas dalam berbagai aplikasi, termasuk komunikasi online yang aman, penyimpanan data, dan perlindungan informasi di perangkat mobile. Ini menggantikan DES (Data Encryption Standard) sebagai standar enkripsi yang lebih kuat dan lebih andal. Keberadaan AES memungkinkan pengguna untuk mengirim dan menyimpan informasi dengan keyakinan bahwa data mereka aman dan terlindungi dari akses yang tidak sah.

AES memiliki tiga varian utama yang berbeda berdasarkan panjang kunci yang digunakan: AES-128, AES-192, dan AES-256. Mari kita bahas setiap varian dengan lebih detail:

1. AES-128:
    - AES-128 menggunakan kunci enkripsi dengan panjang 128 bit.
    - Proses enkripsi menggunakan 10 putaran (rounds) enkripsi.
    - Jumlah putaran ini tergantung pada panjang kunci; AES-128 menggunakan 10 putaran karena kunci yang digunakan memiliki panjang 128 bit.
    - AES-128 menawarkan tingkat keamanan yang tinggi dan efisiensi yang baik dalam penggunaan sumber daya, membuatnya sangat populer dalam banyak aplikasi.
2. AES-192:
    - AES-192 menggunakan kunci enkripsi dengan panjang 192 bit.
    - Proses enkripsi menggunakan 12 putaran.
    - Jumlah putaran ini disesuaikan dengan panjang kunci yang lebih besar, yang memerlukan lebih banyak putaran untuk memastikan keamanan yang setara.
    - AES-192 menawarkan tingkat keamanan yang lebih tinggi daripada AES-128 karena panjang kunci yang lebih besar, tetapi juga sedikit membutuhkan lebih banyak sumber daya.
3. AES-256:
    - AES-256 menggunakan kunci enkripsi dengan panjang 256 bit.
    - Proses enkripsi menggunakan 14 putaran.
    - Panjang kunci yang lebih besar dan jumlah putaran yang lebih banyak membuat AES-256 menjadi varian yang paling kuat dan aman dari AES.
    - AES-256 dapat menghadirkan tingkat keamanan yang sangat tinggi, tetapi pada saat yang sama, memerlukan lebih banyak sumber daya komputasi dibandingkan dengan varian AES yang lebih pendek.

Ketiga varian AES ini memiliki struktur enkripsi yang sama, tetapi menggunakan panjang kunci yang berbeda untuk menyediakan tingkat keamanan yang sesuai dengan kebutuhan aplikasi

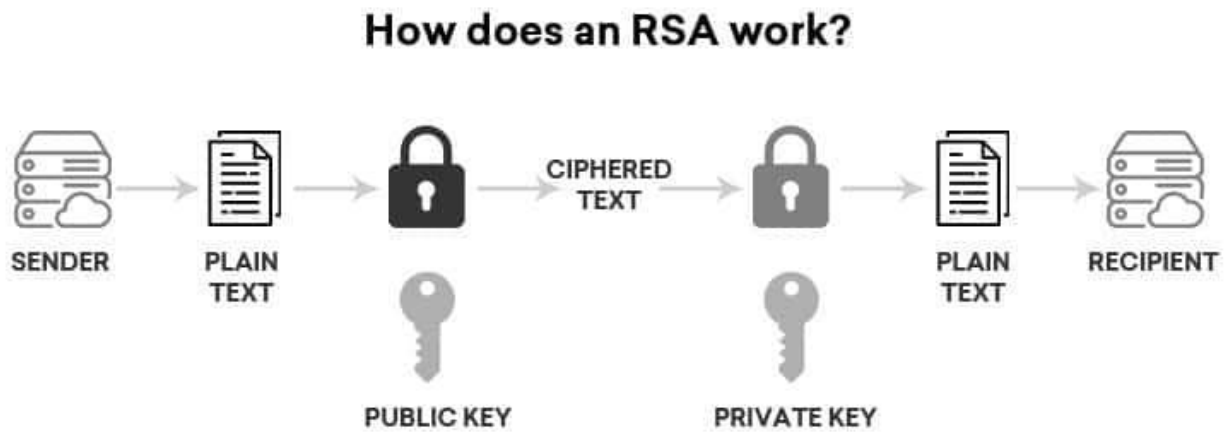
tertentu. Sebagai aturan umum, semakin panjang kunci yang digunakan, semakin sulit bagi penyerang untuk memecahkan enkripsi dan mendapatkan akses ke data yang dilindungi. Oleh karena itu, pemilihan varian AES harus didasarkan pada kebutuhan keamanan spesifik dan ketersediaan sumber daya komputasi yang tersedia.

3.2 RSA (Rivest-Shamir-Adleman)

Algoritma enkripsi Rivest-Shamir-Adleman (RSA) merupakan algoritma enkripsi asimetris yang banyak digunakan di banyak produk dan layanan. Enkripsi asimetris menggunakan pasangan kunci yang secara matematis terhubung untuk mengenkripsi dan mendekripsi data. Kunci privat dan kunci publik dibuat, dengan kunci publik dapat diakses oleh siapa saja dan kunci privat menjadi rahasia yang hanya diketahui oleh pembuat pasangan kunci. Dengan RSA, kunci privat atau publik dapat mengenkripsi data, sementara kunci lainnya mendekripsinya. Inilah salah satu alasan RSA menjadi algoritma enkripsi asimetris yang paling banyak digunakan.

Cara Kerja

Pilihan untuk mengenkripsi dengan kunci privat atau publik menyediakan banyak layanan bagi pengguna RSA. Jika kunci publik digunakan untuk enkripsi, maka kunci privat harus digunakan untuk mendekripsi data. Ini sempurna untuk mengirimkan informasi sensitif melalui jaringan atau koneksi Internet, di mana penerima data mengirimkan kunci publiknya kepada pengirim data. Pengirim data kemudian mengenkripsi informasi sensitif dengan kunci publik dan mengirimkannya ke penerima. Karena kunci publik mengenkripsi data, hanya pemilik kunci



privat yang dapat mendekripsi data sensitif. Dengan demikian, hanya penerima data yang dituju yang dapat mendekripsinya, meskipun data tersebut diambil dalam perjalanan.

Metode enkripsi asimetris lainnya dengan RSA adalah mengenkripsi pesan dengan kunci pribadi. Dalam contoh ini, pengirim data mengenkripsi data dengan kunci pribadinya dan mengirimkan data terenkripsi serta kunci publiknya ke penerima data. Penerima data kemudian dapat mendekripsi data dengan kunci publik pengirim, sehingga memverifikasi bahwa pengirim adalah siapa yang mereka katakan. Dengan metode ini, data dapat dicuri dan dibaca saat transit, namun tujuan sebenarnya dari enkripsi jenis ini adalah untuk membuktikan identitas pengirim. Jika data dicuri dan diubah saat transit, kunci publik tidak akan dapat mendekripsi pesan baru, sehingga penerima akan mengetahui bahwa data telah diubah saat transit.

Detail Teknis RSA

Detail teknis RSA didasarkan pada gagasan bahwa mudah untuk menghasilkan suatu bilangan dengan mengalikan dua bilangan yang cukup besar, namun memfaktorkan bilangan tersebut kembali ke bilangan prima aslinya sangatlah sulit. Kunci publik dan privat dibuat dengan dua bilangan, salah satunya merupakan hasil perkalian dua bilangan prima besar. Keduanya menggunakan dua bilangan prima yang sama untuk menghitung nilainya. Kunci RSA cenderung memiliki panjang 1024 atau 2048 bit, sehingga sangat sulit untuk difaktorkan, meskipun kunci 1024 bit diyakini akan segera dapat dipecahkan.

Kelemahan RSA

Meskipun dapat digunakan dalam banyak situasi, masih ada sejumlah kerentanan di RSA yang dapat dieksploitasi oleh penyerang. Salah satu kerentanan tersebut adalah penerapan kunci panjang dalam algoritma enkripsi. Algoritma seperti AES tidak dapat dipecahkan, sedangkan RSA mengandalkan ukuran kuncinya agar sulit dipecahkan. Semakin panjang kunci RSA, semakin aman kunci tersebut. Dengan menggunakan faktorisasi prima, para peneliti berhasil memecahkan algoritma RSA kunci 768 bit, namun membutuhkan waktu 2 tahun, ribuan jam kerja, dan jumlah daya komputasi yang tidak masuk akal, sehingga panjang kunci yang saat ini digunakan di RSA masih aman. Institut Sains dan Teknologi Nasional (NIST) saat ini merekomendasikan panjang kunci minimum 2048 bit, namun banyak organisasi telah menggunakan kunci dengan panjang 4096 bit. Rentannya RSA lainnya adalah:

Penghasil Angka Acak Lemah

Ketika organisasi menggunakan generator bilangan acak yang lemah, maka bilangan prima yang mereka buat akan lebih mudah untuk difaktorkan, sehingga memudahkan penyerang untuk memecahkan algoritme.

Pembangkitan Kunci yang Lemah

Kunci RSA memiliki persyaratan tertentu terkait dengan pembuatannya. Jika bilangan prima terlalu dekat, atau jika salah satu bilangan penyusun kunci privat terlalu kecil, maka kunci tersebut dapat diselesaikan dengan lebih mudah.

Serangan Saluran Samping

Serangan saluran samping adalah metode serangan yang memanfaatkan sistem yang menjalankan algoritma enkripsi, bukan algoritma itu sendiri. Penyerang dapat menganalisis kekuatan yang digunakan, menggunakan analisis prediksi cabang, atau menggunakan serangan waktu untuk menemukan cara memastikan kunci yang digunakan dalam algoritma, sehingga membahayakan data.

3.3 SHA (Secure Hash Algorithm)

SHA (Secure Hash Algorithm): Merupakan serangkaian algoritma kriptografi yang digunakan untuk menghasilkan nilai hash atau checksum dari data input. Ini berarti bahwa ketika data dimasukkan ke dalam algoritma SHA, hasilnya adalah serangkaian angka dan huruf yang unik untuk setiap input. Dalam konteks keamanan jaringan, ini berfungsi sebagai "sidik jari" unik untuk setiap data.

Unik dan Signifikan: Nilai hash yang dihasilkan oleh SHA bersifat unik, yang berarti setiap perubahan bahkan pada satu bit pun dalam data input akan menghasilkan nilai hash yang sangat berbeda. Hal ini memungkinkan penggunaan nilai hash untuk memverifikasi apakah data telah berubah atau tidak selama proses transfer.

Peran Penting Hashing dalam Keamanan Jaringan

Verifikasi Integritas Data: Fungsi utama SHA dalam keamanan jaringan adalah untuk memastikan integritas data. Ini dilakukan dengan menghasilkan nilai hash dari data yang akan ditransfer dan membandingkannya dengan nilai hash yang dihasilkan setelah data tiba di tujuan. Jika nilai hash berbeda, ini menunjukkan bahwa data telah diubah selama proses transfer, yang dapat menandakan adanya serangan atau manipulasi.

Penggunaan dalam Protokol Keamanan: SHA sering digunakan dalam protokol keamanan seperti TLS (Transport Layer Security) dan HTTPS (Hypertext Transfer Protocol Secure) untuk mengamankan komunikasi data antara server dan klien. Dalam konteks ini, SHA digunakan untuk menghasilkan tanda tangan digital yang memverifikasi integritas data yang ditransfer.

Versi SHA dan Tingkat Keamanan

Berbagai Versi: Ada beberapa versi dari SHA, termasuk SHA-1, SHA-256, SHA-384, dan SHA-512, yang masing-masing memiliki panjang output hash yang berbeda.

Tingkat Keamanan: Versi yang lebih baru, seperti SHA-256, menawarkan tingkat keamanan yang lebih tinggi daripada versi yang lebih lama, seperti SHA-1. Ini karena algoritma hashing yang lebih baru cenderung memiliki kerumitan yang lebih tinggi, sehingga lebih sulit untuk diretas atau dimanipulasi oleh penyerang.

3.1.1 Aplikasi SHA dalam Keamanan Jaringan

Penggunaan Luas: SHA digunakan dalam berbagai protokol keamanan seperti TLS, HTTPS, DNSSEC (Domain Name System Security Extensions), dan lainnya.

Verifikasi Integritas: Dalam konteks keamanan jaringan, SHA memastikan bahwa data yang ditransfer antara komputer tidak diubah atau dimanipulasi oleh pihak yang tidak berwenang, yang merupakan aspek penting dari keamanan informasi.

3.1.2 Keamanan dan Perkembangan Terbaru

Evaluasi Terus Menerus: Meskipun SHA-256 saat ini dianggap cukup aman, tidak ada jaminan bahwa itu akan tetap aman di masa depan. Oleh karena itu, para ahli keamanan terus memantau perkembangan dalam kriptografi untuk mengidentifikasi dan menangani ancaman keamanan baru.

Perlunya Kepatuhan: Penting bagi organisasi untuk mematuhi praktik keamanan terbaik yang diberlakukan oleh badan-badan standar dan regulasi, serta untuk selalu menggunakan versi terbaru dan teraman dari algoritma hashing seperti SHA.

3.2 Fungsi Hash

Fungsi hash kriptografis adalah salah satu dari sekumpulan fungsi hash yang cocok untuk aplikasi kriptografi seperti SSL /TLS. Seperti fungsi hash lainnya, fungsi hash kriptografis adalah algoritma matematika satu arah yang digunakan untuk memetakan data ukuran apa saja ke string bit dari ukuran tetap. Fungsi hash kriptografis banyak digunakan dalam praktik keamanan informasi, seperti tanda tangan digital, kode otentikasi pesan, dan bentuk otentikasi lainnya.

Fungsi hash kriptografi harus memiliki properti berikut (sumber: wikipedia):

1. Pesan yang sama selalu menghasilkan nilai hash yang sama (yaitu fungsinya deterministik).
2. Nilai hash dihitung dengan cepat.
3. Tidak mungkin memiliki dua pesan dengan nilai hash yang sama (dikenal sebagai "collision").
4. Tidak mungkin secara sengaja membuat pesan yang menghasilkan nilai hash tertentu.
5. Sedikit perubahan pada pesan harus mengubah nilai hash yang dihasilkan secara luas, sehingga tampaknya tidak berkorelasi dengan hash asli.

Fungsi hash kriptografi yang paling umum digunakan MD5, SHA-1, dan SHA-2. Keunikan masing-masing hash sangat penting untuk integritas fungsi hash kriptografi. Inilah yang benar-benar membedakan fungsi hash kriptografis dari fungsi hash lainnya - kepastian bahwa pesan tertentu diidentifikasi dengan cara yang unik dan tidak dapat digandakan.

Skema tanda tangan digital (seperti untuk penandatanganan dokumen, penandatanganan kode, atau S/MIME e-mail) umumnya mensyaratkan bahwa hash kriptografi dihitung dari pesan dan dimasukkan dalam tanda tangan. Perangkat lunak penerima kemudian secara independen menghitung hash untuk memverifikasi integritas pesan.

Situs web juga sering menerbitkan nilai hash untuk file yang dapat diunduh. Ketika pengguna mengunduh file, mereka dapat menggunakan perangkat lunak mereka sendiri untuk secara independen menghitung hash, memverifikasi integritas file.

Keamanan kata sandi juga bergantung pada hash kriptografi. Kata sandi yang disajikan oleh pengguna di-hash dan kemudian dibandingkan dengan hash yang disimpan.

Fungsi hash kriptografis banyak digunakan dalam protokol keamanan seperti SSL /TLS dan SSH, dan dalam aplikasi lain yang mengandalkan integritas data. Cryptocurrency menggunakan algoritma hashing untuk memperbarui blockchain dengan blok baru data transaksi yang aman dan dapat diverifikasi. (BitCoin, misalnya, menggunakan SHA-2 untuk verifikasi transaksi.)

BAB 4
PROTOKOL KRIPTOGRAFI

Protokol kriptografi adalah sebuah protokol abstrak atau konkret yang melakukan fungsi terkait keamanan dan menerapkan metode kriptografi, sering sebagai urutan primitif kriptografi. Protokol yang menjelaskan bagaimana algoritma harus digunakan. Sebuah protokol yang cukup rinci mencakup rincian tentang struktur data dan representasi, di mana titik itu dapat digunakan untuk mengimplementasikan beberapa, versi dioperasikan dari sebuah program. Protokol kriptografi banyak digunakan untuk pengiriman data tingkat aplikasi yang aman. Protokol kriptografi biasanya menyertakan setidaknya beberapa aspek berikut: Protokol kriptografi banyak digunakan untuk pengiriman data tingkat aplikasi yang aman. Protokol kriptografi biasanya menyertakan setidaknya beberapa aspek berikut:

- Perjanjian kunci atau pendirian
- Otentikasi entitas
- Simetris enkripsi dan otentikasi pesan bahan konstruksi
- Aman tingkat aplikasi data transportasi
- Metode non-repudiation
- Metode berbagi rahasia
- Komputasi multi-pihak yang aman

Misalnya, Transport Layer Security (TLS) adalah protokol kriptografi yang digunakan untuk mengamankan sambungan web (HTTP/HTTPS). Ini memiliki mekanisme otentikasi entitas, berdasarkan sistem X. 509; fase pengaturan kunci, di mana kunci enkripsi simetris terbentuk dengan menggunakan kriptografi kunci publik; dan fungsi transpor data tingkat aplikasi. Ketiga aspek ini memiliki interkoneksi penting. Standar TLS tidak memiliki dukungan non-repudiasi.

Ada jenis lain dari protokol kriptografi juga, dan bahkan istilah itu sendiri memiliki berbagai bacaan; Protokol Aplikasi kriptografi sering menggunakan satu atau lebih metode perjanjian kunci yang mendasari, yang juga terkadang sendiri disebut sebagai "protokol kriptografi ". Misalnya, TLS mempekerjakan apa yang dikenal sebagai pertukaran kunci Diffie-Hellman, yang meskipun hanya bagian dari TLS per se, Diffie-Hellman dapat dilihat sebagai protokol kriptografi lengkap dalam dirinya sendiri untuk aplikasi lain.

4.1 HTTPS (Hypertext Transfer Protocol Secure)

Hypertext Transfer Protocol Secure (HTTPS) adalah ekstensi dari Hypertext Transfer Protocol (HTTP). Ini digunakan untuk komunikasi aman melalui jaringan komputer, dan banyak digunakan di Internet. Dalam HTTPS, protokol komunikasi dienkripsi menggunakan Transport Layer Security (TLS) atau, sebelumnya, Secure Sockets Layer (SSL). Oleh karena itu, protokol ini juga disebut sebagai HTTP over TLS, atau HTTP over SSL.

Motivasi utama untuk HTTPS adalah otentikasi situs web yang diakses, dan perlindungan privasi dan integritas data yang dipertukarkan saat dalam proses. Ini melindungi terhadap serangan man-in-the-middle, dan enkripsi komunikasi dua arah antara klien dan server melindungi komunikasi terhadap penyadapan dan gangguan. Dalam praktiknya, ini memberikan jaminan yang masuk akal bahwa seseorang berkomunikasi dengan situs web yang dimaksud tanpa gangguan dari penyerang.

Aspek otentikasi HTTPS mengharuskan pihak ketiga tepercaya untuk menandatangani sertifikat digital sisi-server. Ini secara historis merupakan operasi yang mahal, yang berarti koneksi HTTPS yang sepenuhnya terautentikasi biasanya hanya ditemukan pada layanan transaksi pembayaran aman dan sistem informasi perusahaan aman lainnya di World Wide Web. Pada 2016, kampanye oleh Electronic Frontier Foundation dengan dukungan pengembang web browser menyebabkan protokol menjadi lebih lazim.

4.1.1 Penggunaan situs web

Pada April 2018, 33,2% dari 1.000.000 situs web teratas Alexa menggunakan HTTPS sebagai default, 57,1% dari 137.971 situs web paling populer di Internet memiliki implementasi HTTPS yang aman, dan 70% dari pemuatan halaman (diukur oleh Firefox Telemetry) menggunakan HTTPS.

4.1.2 Integrasi peramban

Sebagian besar peramban menampilkan peringatan jika mereka menerima sertifikat yang tidak valid. Peramban yang lebih lama, saat menghubungkan ke situs dengan sertifikat yang tidak valid, akan menghadirkan kotak dialog kepada pengguna yang menanyakan apakah mereka ingin melanjutkan. Peramban yang lebih baru menampilkan peringatan di seluruh jendela. Peramban

yang lebih baru juga secara jelas menampilkan informasi keamanan situs di bilah alamat. Sertifikat validasi diperpanjang mengubah bilah alamat menjadi hijau di peramban yang lebih baru. Sebagian besar peramban juga menampilkan peringatan kepada pengguna saat mengunjungi situs yang berisi campuran konten terenkripsi dan tidak terenkripsi. Selain itu, banyak filter web mengembalikan peringatan keamanan saat mengunjungi situs web terlarang.

Electronic Frontier Foundation, berpendapat bahwa "Di dunia yang ideal, setiap permintaan web dapat default ke HTTPS", telah menyediakan add-on yang disebut HTTPS Everywhere untuk Mozilla Firefox, Google Chrome, Chromium, dan Android, yang memungkinkan HTTPS secara default untuk ratusan situs web yang sering digunakan.

4.1.3 Perbedaan Dengan HTTP

URL HTTPS dimulai dengan "https://" dan menggunakan port 443 secara default, sedangkan, HTTP URL dimulai dengan "http://" dan gunakan port 80 secara default. HTTP tidak dienkripsi dan karenanya rentan terhadap serangan man-in-the-middle dan penyadapan, yang dapat memungkinkan penyerang mendapatkan akses ke akun situs web dan informasi sensitif, dan memodifikasi halaman web untuk menyuntikkan malware atau iklan. HTTPS dirancang untuk menahan serangan semacam itu dan dianggap aman terhadap serangan itu (dengan pengecualian implementasi HTTPS yang menggunakan versi SSL yang sudah tidak digunakan lagi).

4.1.4 Pengaturan Server

Untuk menyiapkan server web untuk menerima koneksi HTTPS, administrator harus membuat sertifikat kunci publik untuk server web. Sertifikat ini harus ditandatangani oleh otoritas sertifikat tepercaya agar peramban web dapat menerimanya tanpa peringatan. Otoritas menyatakan bahwa pemegang sertifikat adalah operator dari server web yang menyajikannya. Peramban web umumnya didistribusikan dengan daftar sertifikat penandatanganan otoritas sertifikat utama sehingga mereka dapat memverifikasi sertifikat yang ditandatangani oleh mereka.

4.2 TLS/SSL (Transport Layer Security/Secure Sockets Layer)

Keamanan Lapisan Transportasi Inggris: Transport Layer Security (TLS), dan pendahulunya yang sudah usang, Secure Sockets Layer (SSL),[1] adalah protokol kriptografi yang dirancang untuk memberikan keamanan komunikasi melalui jaringan komputer.[2] Beberapa versi protokol TLS dapat ditemukan penerapannya secara luas seperti

di peramban web, surel, pesan instan, dan voice over IP (VoIP). Situs web dapat menggunakan TLS untuk mengamankan semua komunikasi antara peladen dan peramban web.

Protokol TLS bertujuan terutama untuk memberikan privasi dan integritas data antara dua atau lebih aplikasi komputer yang berkomunikasi.[2] Ketika diamankan oleh TLS, koneksi antara klien (misalnya, peramban web) dan peladen (misalnya, wikipedia.org) harus memiliki satu atau beberapa properti berikut:

- Koneksi bersifat pribadi (atau aman) karena kriptografi simetris digunakan untuk mengenkripsi data yang dikirimkan. Kunci untuk enkripsi simetris ini dihasilkan secara unik untuk setiap koneksi dan didasarkan pada rahasia bersama yang dinegosiasikan pada awal sesi. Peladen dan klien menegosiasikan perincian algoritma enkripsi dan kunci kriptografi mana yang digunakan sebelum bita pertama data ditransmisikan. Negosiasi rahasia bersama aman (rahasia yang dinegosiasikan tidak tersedia untuk penyadap dan tidak dapat diperoleh, bahkan oleh penyerang yang menempatkan diri di tengah-tengah koneksi) dan dapat diandalkan (tidak ada penyerang dapat memodifikasi komunikasi selama negosiasi tanpa menjadi terdeteksi).
- Identitas pihak yang berkomunikasi dapat diautentikasi menggunakan kunci kriptografi publik. Otentikasi ini dapat dibuat opsional, tetapi umumnya diperlukan untuk setidaknya salah satu pihak (biasanya peladen).
- Koneksi ini dapat diandalkan karena setiap pesan yang dikirim mencakup pemeriksaan integritas pesan menggunakan kode otentikasi pesan untuk mencegah kehilangan atau perubahan data yang tidak terdeteksi selama transmisi.

Selain properti di atas, konfigurasi TLS yang hati-hati dapat memberikan properti terkait privasi tambahan seperti forward secrecy, memastikan bahwa setiap pengungkapan kunci enkripsi di masa depan tidak dapat digunakan untuk mendekripsi setiap komunikasi TLS yang direkam yang sebelumnya.

4.2.1 Secure Sockets Layer (SSL)

SSL adalah protokol keamanan yang digunakan untuk mengamankan komunikasi data antara dua sistem yang terhubung melalui internet. Dikembangkan oleh Netscape pada tahun 1990-an,

SSL berfungsi untuk menyediakan lapisan keamanan tambahan di atas protokol TCP/IP yang mendasari.

Fungsi utama SSL adalah untuk mengenkripsi data yang ditransfer antara klien (seperti browser web) dan server. Ini dilakukan menggunakan kunci enkripsi yang dihasilkan selama proses pertukaran kunci SSL.

Proses Kerja SSL adalah ketika klien menginisiasi koneksi ke server yang dilindungi SSL, keduanya akan melakukan serangkaian tindakan pertukaran kunci enkripsi dan otentikasi untuk memastikan keamanan komunikasi. Ini melibatkan penggunaan sertifikat digital yang dikeluarkan oleh otoritas sertifikasi (CA) untuk memverifikasi identitas server.

Penggunaan umum SSL sering digunakan untuk mengamankan transaksi online, seperti pembelian e-commerce, login ke akun online, dan transfer data sensitif lainnya. Ini juga digunakan untuk mengamankan komunikasi email dan akses ke server web yang dilindungi.

4.2.2 Transport Layer Security (TLS)

TLS adalah standar keamanan yang menggantikan SSL dan digunakan untuk melindungi privasi dan integritas data dalam komunikasi melalui jaringan komputer, termasuk internet. TLS dikembangkan sebagai perbaikan terhadap SSL dengan memperbaiki sejumlah kerentanan keamanan yang ditemukan dalam versi SSL sebelumnya.

Pengembangan dari SSL: TLS sebenarnya merupakan pengembangan dari SSL. Versi pertama TLS, yaitu TLS 1.0, awalnya dikenal sebagai SSL 3.1. Sejak itu, TLS telah berkembang menjadi beberapa versi, termasuk TLS 1.1, TLS 1.2, dan yang terbaru, TLS 1.3.

Fungsi dan Keamanan: TLS memiliki fungsi yang mirip dengan SSL dalam hal mengenkripsi dan melindungi data selama komunikasi. Namun, TLS juga menawarkan peningkatan keamanan dan efisiensi dibandingkan dengan versi SSL sebelumnya. Misalnya, TLS 1.3 memperkenalkan peningkatan signifikan dalam hal keamanan, privasi, dan efisiensi dengan mengurangi jumlah ronde tangan yang diperlukan selama proses pertukaran kunci.

Proses Koneksi: Seperti SSL, TLS juga melibatkan serangkaian pertukaran kunci enkripsi dan otentikasi antara klien dan server untuk memastikan keamanan komunikasi. Ini termasuk

verifikasi sertifikat digital oleh klien untuk memverifikasi identitas server dan mendukung berbagai algoritma enkripsi dan hash untuk menyediakan lapisan keamanan yang kuat.

4.2.3 Perbandingan Antara SSL dan TLS

Keamanan: Secara umum, TLS dianggap lebih aman daripada SSL karena memperbaiki sejumlah kerentanan yang ada dalam versi SSL sebelumnya. Misalnya, TLS 1.3 memperkenalkan peningkatan keamanan dengan menghapus dukungan untuk kriptografi yang sudah tidak aman dan meningkatkan efisiensi koneksi.

Kompatibilitas: Meskipun TLS telah menjadi standar keamanan yang disarankan, beberapa sistem dan perangkat lunak masih mendukung SSL untuk kompatibilitas mundur. Namun, secara bertahap, banyak organisasi telah beralih dari SSL ke TLS sebagai standar keamanan yang lebih kuat.

Peningkatan Performa: Beberapa versi TLS, seperti TLS 1.2 dan TLS 1.3, telah memperkenalkan peningkatan dalam hal keamanan, privasi, dan efisiensi, menjadikannya pilihan yang lebih baik daripada versi SSL yang lebih lama. Misalnya, TLS 1.3 mengurangi latensi dan jumlah ronde tangan yang diperlukan selama proses pertukaran kunci, meningkatkan kinerja secara keseluruhan.

Adopsi Industri: Secara bertahap, industri telah beralih dari SSL ke TLS sebagai standar keamanan untuk melindungi komunikasi internet. Organisasi dan penyedia layanan web di seluruh dunia telah mengadopsi TLS untuk menyediakan lapisan keamanan tambahan bagi pengguna mereka.