BAB II
TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Sampai saat ini sudah banyak sistem keamanan *database* yang sudah dikembangan dan dituangkan dalam bentuk penelitian dengan berbagai macam metode yang digunakan. Referensi ini digunakan sebagai dasar dalam penyusunan penelitian ini yang memiliki keterkaitan dengan objek maupun metode penelitian yang diangkat. Referensi yang dikutip juga dijadikan landasan mengenai batasan-batasan yang ditetapkan dalam penelitian ini. Ada berbagai macam penelitian yang mengangkat topik sistem keamanan *database* sejauh ini.

Yeni Yanti dkk mengimplementasikan algoritma *Triangle Chain* untuk sistem yang dirancang. Penelitian ini dilakukan untuk menerapkan algoritma enkripsi *Triangle Chain* pada sistem keamanan *database* yang dirancang. Pengujian dilakukan dengan objek *database* berisi *file* teks normal tanpa format. Sistem dibagun menggunakan aplikasi berbasis *Visual Basic* dengan hasil aplikasi berbasis *Visual Basic* dapat digunakan sebagai pengaman *database* [11].

Nurhikmah Taliasih dan Irawan Afrianto dalam penelitiannya berjudul "**Sistem Keamanan Basis Data Klien P.T. Infokes Menggunakan Kriptografi Kombinasi RC4 Dan Base64**" menerapkan algoritma enkripsi RC4 pada sistem yang dibagun dengan kombinasi Base64. Pengujian dilakukan menggunakan *black box* dan *white box* serta *Wireshark* untuk mengetahui apakah enkripsi melalui protokol HTTPS sudah bekerja atau belum. Sementara hasil pengujian aspek kriptografi dengan metode kripanalisis menggunakan CrypTool menunjukkan jika kombinasi RC4 dan Base64 memiliki tingkat keamanan yang lebih tinggi dibandingkan hanya menggunakan RC4 [12].

Susanto dalam penelitiannya memaparkan bagaimana implementasi algoritma *Blowfish* dalam sistem keamanan basis data. Penelitian dilakukan dengan studi kasus sistem informasi koperasi RIAS. Pada sistem informasi yang dibangun berbasis bahasa pemrograman PHP dan *database* MySQL, diterapkan algoritma *Blowfish* dimana *public key* diinputkan ke dalam *source code* pada sistem

informasinya itu sendiri. Hasil pengujian menunjukkan data yang diinputkan berhasil dienkripsi sehingga tidak terbaca pada sistem informasi koperasi RIAS [13].

Peneliti Putra Rahmadi dan Hilda Dwi Yunita melakukan implementasi enkripsi pada sistem keamanan *database* dengan menggunakan metode enkripsi *stream cipher*. Dalam penelitian dengan studi kasus PT. Sugar Group Companies tersebut, didapatkan hasil sebuah program yang dapat melakukan enkripsi dan dekripsi pada basis data PT. Sugar Group Companies sehingga mengurangi risiko pencurian data atau kebocoran data oleh pihak yang tidak bertanggung jawab [14].

Sementara kombinasi RSA dan AES sendiri sudah diimplementasikan oleh peneliti Aditya Hermawan dan Erik Iman Heri Ujianto (2021) sebagai basis enkripsi data. Dalam penelitian yang mengembangkan aplikasi enkripsi dan dekripsi tersebut, didapatkan hasil waktu pembangkitan kunci 2048 bit dengan waktu *create* 495,56 milidetik. Hasil juga menunjukkan jika enkripsi dengan RSA membutuhkan waktu rata-rata 7.428 milidetik, dan enkripsi dengan AES membutuhkan waktu rata-rata 4.365 milidetik. Proses dekripsi menggunakan RSA rata-rata memakan waktu 54.068 milidetik, sedangkan proses dekripsi menggunakan AES rata-rata memakan waktu 2.753 milidetik [15].

Enkripsi kombinasi dua algoritma dilakukan oleh peneliti Rian Oktafiani, Erik Iman Heri Ujianto, dan Rianto dalam penelitian yang berjudul "**Kombinasi Algoritma Kriptografi Vigenere Cipher dan SHA256 untuk Keamanan Basis Data**". Pada penelitian yang menggunakan database pada pelayanan administrasi desa Girisuko tersebut, enkripsi Vigenere dan SHA256 diterapkan secara bersamaan sehingga menghasilkan *hash value* yang sulit untuk dipecahkan. Penerapan Vigenere Cipher dan SHA pada aplikasi sistem informasi manajemen desa dengan basis data real-time berjalan dengan baik, ditunjukkan dengan runtime yang cepat 0,39 detik, proses enkripsi data menggunakan enkripsi Vigenere dengan hasil 894.968 kunci per detik, dan dianalisis. panjang kunci 7 karakter dan teks pada *database* penduduk dapat diamankan [16].

Masih melibatkan *Vigenere Cipher*, peneliti Muhammad Apit Ruswandi dan Windarto mengkombinasikannya dengan AES-128. Dalam penelitian berjudul

"**Enkripsi Database Sistem Informasi Helpdesk Dengan Algoritme Kriptografi AES-128 dan Vigenere Chiper**" tersebut, algoritma AES-128 dan *Vigenere* berhasil diimplementasikan dalam aplikasi help desk yang dikembangkan menggunakan bahasa pemrograman PHP dan database MySQL untuk mengamankan *database* di kolom tertentu yang telah ditentukan [17].

Sementara penelitian yang dilakukan Muhammad Andrian Nasuton, Edy Victor Haryanto dan Alfa Saleh yakni menggabungkan *Hill Cipher* dan *Stream Cipher* untuk keamanan *database MySQL*. Dalam sistem yang dibangun menggunakan *Visual Basic* 2010 tersebut, sistem dapat mengamankan data pada *database MySQL* dengan baik menggunakan kombinasi enkripsi *Hill* dan *Stream Cipher* [18].

Penelitian oleh Annas Rifa'I dan Lilis Cucu Sumartini dengan judul penelitian "**Implementasi Kriptografi Menggunakan Metode Blowfish Dan Base 64 Untuk Mengamankan Database Informasi Akademik Pada Kampus Akademi Telekomunikasi Bogor Berbasis Web-Based**". Kombinasi dari Base64 dan Blowfish berhasil diimplementasikan pada *database* informasi Akademi Telekomunikasi Bogor dalam sistem berbasis *Web Based* [19].

Sementara itu, penelitian oleh Ahsanul Rahman dan Sri Mulyati dalam penelitiannya yang berjudul "**Implementasi Keamanan Database Menggunakan Algoritma Vigenere Cipher Dan Rivest Shamir Adleman (RSA) Berbasis Desktop**" berhasil membangun aplikasi desktop untuk mengamankan database menggunakan kombinasi enkripsi *Vigenere Cipher* dan RSA. Hasil penelitian menunjukan jika aplikasi pengaman *database* yang dirancang dapat melakukan enkripsi dan dekripsi pada isi *database* menggunakan kombinasi *Vigenere Cipher* dan RSA [20].

Tabel 2. 1 Penelitian Terdahulu

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
Yani Yati, Munawir, Zulfan, Erdiwansyah	Implementasi Sistem Keamanan Database Menggunakan Metode *Triangle Chain* [11].	Enkripsi Triange Chain	Implementasi dari algoritma *Triangle Chain* dengan hasil yang didapatkan yakni aplikasi sistem keamanan yang dibuat menggunakan *visual basic* dapat diimplementasikan untuk mengamankan sistem informasi basis data.	Perancangan sistem menggunakan *visual basic*, bukan menggunakan sistem model web hosting.
Nurhikmah Taliasih, Irawan Afrianto	Sistem Keamanan Basis Data Klien P.T. Infokes Menggunakan Kriptografi Kombinasi	Enkripsi RC4 Dan *Base64*	Kombinasi dari dua enkripsi yakni RC4 dan Base64 yang dapat diimplementasikan	Fokus pengujian pada aspek tingkat keamanan ketika menggunakan salah satunya saja dan

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
	RC4 Dan *Base64* [12].		dalam penelitian tersebut dengan hasil penelitian pengkombinasian kedua algoritma tersebut menghasilkan tingkat keamanan yang lebih tinggi dibandingkan dengan hanya menggunakan salah satunya saja dalam mengamankan basis data.	keduanya dan bukan pada aspek kombinasi enkripsinya saja.
Susanto	Implementasi Keamanan Data Menggunakan Algoritma Blowfish Pada Sistem Informasi Koperasi RIAS [13].	Blowfish	Penelitian ini berfokus pada penerapan algortima *Blowfish* pada sebuah studi kasus sistem informasi	Penelitian berfokus pada studi kasus Sistem Informasi Koperasi RIAS dan bukan perancangan sistem

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
			koperasi RIAS dengan perancangan sistem informasi berbasis *php* dan *database MySQL* dengan hasil data yang diinput pada sistem informasi koperasi RIAS dapat dienkripsi dengan mengimplementasikan algoritma *Blowfish*.	yang sepenuhnya baru.
Putra Rahmadi, Hilda Dwi Yunita	Implementasi Pengamanan Basis Data Dengan Teknik Enkripsi (Studi Kasus : PT . Sugar Group Companies) [14].	Stream Cipher	Implementasi metode *stream cipher* yang mana merupakan algortima enkripsi simetris dimana data enkripsi memiliki ukuran yang sama dengan data asli	Penelitian berfokus pada studi kasus PT . Sugar Group Companies dan bukan perancangan sistem yang sepenuhnya baru.

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
			(*plaintext*) dengan hasil penelitian yakni berhasil diterapkannya *stream cipher* pada studi kasus yang diambil yakni PT. Sugar Group Companies sehingga meminimalisir adanya pencurian dan kebocoran data pada basis data.	
Aditya Hermawan, Erik Iman Heri Ujianto	Implementasi Enkripsi Data Menggunakan Kombinasi AES dan RSA [15].	AES dan RSA	Untuk pembuatan kunci ideal yakni menggunakan 2048 bit, dan waktu pembuatan diperkirakan sekitar 495,56 milidetik. Perlu disebutkan bahwa proses dekripsi membutuhkan	Fokus penelitian pada perancangan aplikasi untuk keamanan data dan tidak berfokus pada suatu sistem web yang terimplementasi AES dan RSA.

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
			waktu yang relatif lebih lama daripada proses enkripsi. Waktu yang dibutuhkan oleh RSA untuk enkripsi adalah sekitar 7,428 milidetik, sedangkan untuk AES sekitar 4,365 milidetik. Sebaliknya, dekripsi RSA diperkirakan membutuhkan waktu rata-rata 54,068 milidetik, dan untuk AES sekitar 2,753 milidetik.	
Rian Oktafiani, Erik Iman Heri Ujianto, dan Rianto	Kombinasi Algoritma Kriptografi Vigenere Cipher dan SHA256	Vigenere Cipher dan SHA256	Penelitian dengan mengkombinasikan Vigenere Cipher dengan	Penelitian menggunakan metode kombinasi *Caesar Cipher* dengan

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
	untuk Keamanan Basis Data		SHA256 pada database sistem layanan administrasi desa Girisuko dengan kedua algortima yang dikombinasikan tersebut menghasilkan data terenkripsi dengan *output hash value* pada *database* data penduduk desa Girisuko.	*hashing* dan tidak menggunakan kombinasi simetris dan asimetris.
Muhammad Apit Ruswandi dan Windarto	Enkripsi Database Sistem Informasi Helpdesk Dengan Algoritme Kriptografi AES-128 dan Vigenere Chiper [17]	AES-128 dan Vigenere	Penelitian kriptografi yang mengkombinasikan enkripsi *Vigenere Cipher* dengan AES-128 pada aplikasi *helpdesk* dengan hasil kombinasi	Penelitian menggunakan AES-128 (bukan AES-256) dan dikombinasikan dengan *Caesar cipher* yaitu *Vigenere* (bukan asimetris seperti RSA)

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
			kedua model enkripsi tersebut dapat diimplementasikan untuk mengamankan data pada *database* sesuai dengan *field* yang telah ditentukan.	
Muhammad Andrian Nasuton, Edy Victor Haryanto dan Alfa Saleh	Penerapan Metode Hill Cipher Dan Stream Cipher Dalam Mengamankan Database MySQL [18]	*Hill Cipher* dan *Stream Cipher*	Penelitian yang berfokus pada perancangan sistem menggunakan *Visual Basic* 2010 dengan mengkombinasikan *Hill Cipher* dan *Stream Cipher* dimana kombinasi keduanya berhasil diterapkan untuk mengamankan data pada *database*	Sama-sama menggunakan tipe database MySQL dengan pembeda terletak pada kombinasi enkripsi yang digunakan (Hill dan Stream Cipher, bukan AES-256 dan RSA)

Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
			MySQL.	
Annas Rifa'I dan Lilis Cucu Sumartini	Implementasi Kriptografi Menggunakan Metode Blowfish Dan Base 64 Untuk Mengamankan Database Informasi Akademik Pada Kampus Akademi Telekomunikasi Bogor Berbasis Web-Based [19]	Blowfish dan Base64	Penelitian kriptografi dengan mengkombinasikan enkripsi Blowfish dan Base64 yang berhasil diimplementasikan pada database informasi akademik Akademi Telekomunikasi Bogor berbasis web.	Studi kasus dengan metode enkripsi dua tahap pada Database Informasi Akademik Pada Kampus Akademi Telekomunikasi Bogor dan tidak menggunakan model perancangan sistem yang sepenuhnya baru.
Ahsanul Rahman dan Sri Mulyati	Implementasi Keamanan Database Menggunakan Algoritma Vigenere Cipher Dan Rivest Shamir Adleman (RSA) Berbasis Desktop [20]	Vigenere dan RSA	Perancangan aplikasi *desktop* untuk mengamankan *database* dengan mengkombinasikan *Vigenere Cipher* dan	Metode kombinasi menggunakan enkripsi Vigenere namun sama-sama menggunakan RSA.

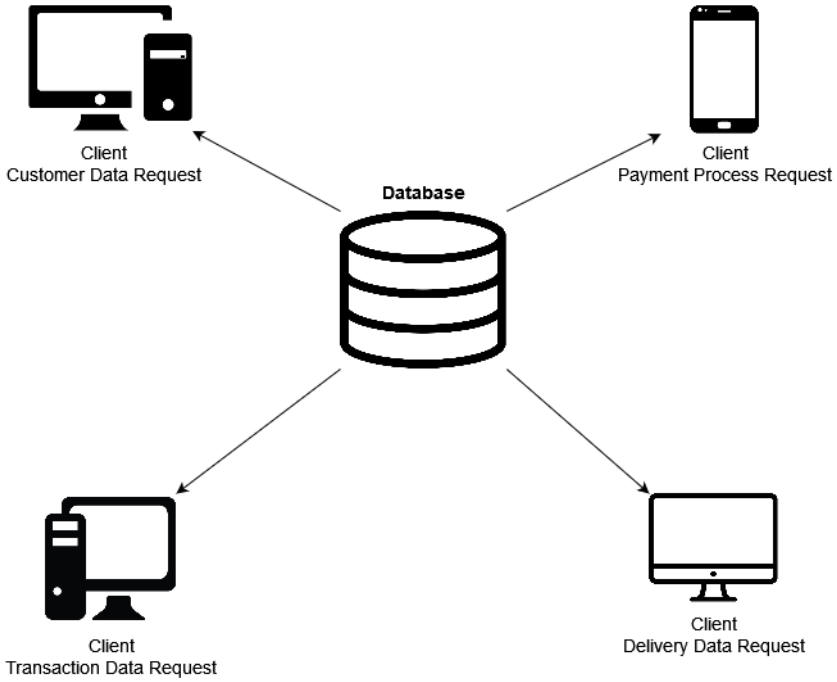
Peneliti	Judul Penelitian	Metode	Hasil	Perbandingan
			RSA dengan hasil penelitian aplikasi *desktop* yang dirancang dapat dijalankan untuk mengamankan *database*.	

Kesimpulan dari beberapa penelitian terdahulu pada tabel diatas adalah algoritma enkripsi yang dapat diterapkan dan diimplementasikan sebagai mekanisme pertahanan terhadap serangan pada basis data. Dari beberapa referensi penelitian terdahulu diatas, perbedaan penelitian sebelumnya yang digunakan sebagai referensi penelitian ini dengan penelitian yang dilakukan oleh peneliti adalah penggunaan dua algoritma kriptografi yaitu *Advanced Encryption Standard* (AES) dan Rivest Shamir Adleman (RSA) sebagai basis dari sistem keamanan basis data yang akan diimplementasikan. Kombinasi kedua model kriptografi tersebut akan dibangun *library OpenSSL* dengan objek penelitian adalah aplikasi web berbasis *PHP Native* yang tersimpan pada *web hosting*.

2.2 Landasan Teori

2.2.1 *Database*

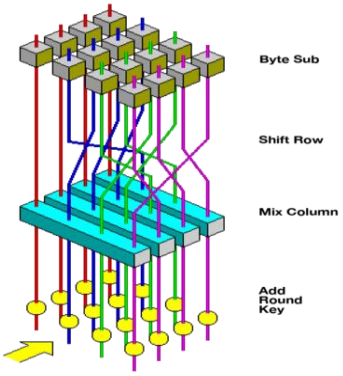
Basis data atau lebih dikenal sebagai *database* adalah tempat penyimpanan data yang terhubung secara *real-time* untuk memenuhi aspek informasi dari suatu organisasi [21]. Basis data berguna untuk organisasi di mana penyimpanan data dan fungsi manajemen adalah hal utama yang dapat diandalkan. Dengan penggunaan *database*, pengelolaan dan akses data akan menjadi lebih cepat dan mudah [22]. Dalam pengertian lain, basis data adalah kumpulan informasi yang disimpan secara sistematis dalam sistem sehingga dapat diperiksa menggunakan program tertentu untuk memperoleh informasi dari basis data [23]. *Database* sendiri memiliki beberapa *platform* yang umum digunakan seperti *MySQL*, *Oracle* dan *PostgreSQL*. Dalam kaitannya dengan penelitian ini, terdapat berbagai macam serangan *database* yang umum terjadi salah satunya adalah *SQL Injection*.



Gambar 2. 1 Ilustrasi Database

Fungsi *database* sendiri diantaranya adalah sebagai penyimpanan data terkait dalam suatu sistem informasi. Kemudahan akses juga dimiliki oleh sistem *database* dimana administrator dapat dengan mudah melakukan CRUD (*Create, Read, Update, Delete*) pada sistem *database* itu sendiri. Selain itu, *database* juga dapat digunakan sebagai penyimpanan data aplikasi dimana banyak aplikasi modern saat ini yang menggunakan sistem penyimpanan *database* untuk menyimpan data. Dari sisi manajemen, adanya klasifikasi pengelompokkan data pada sistem *database* dapat mempermudah administrator dalam mengelola semua data yang tersimpan dalam sistem *database*.

2.2.2 Algoritma *Advanced Encryption Standard* (AES)



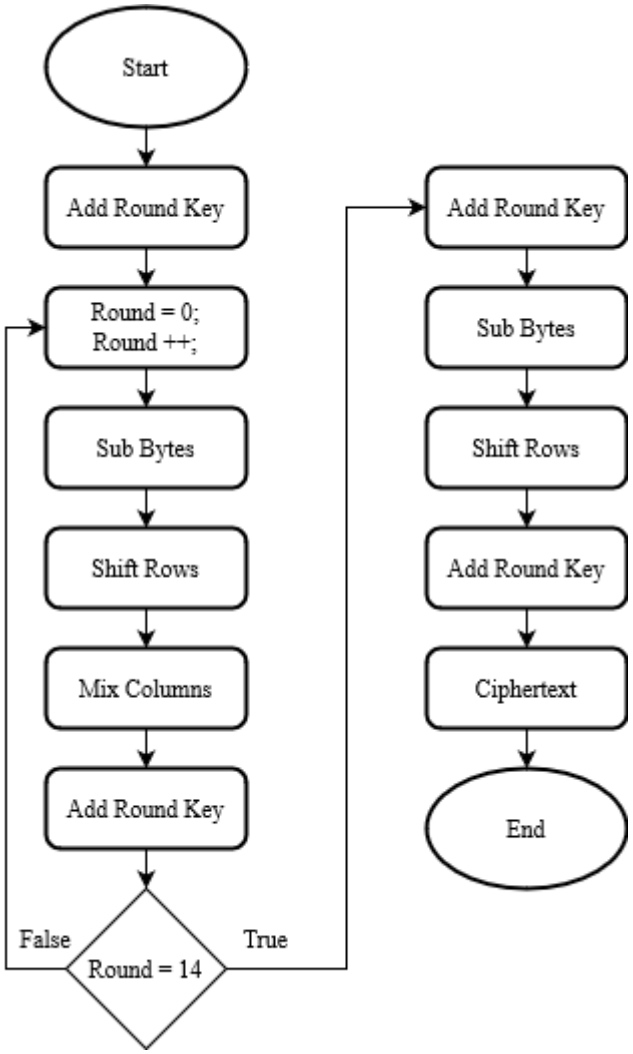
Gambar 2. 2 Alur Kerja Algoritma AES

Algoritma *Advanced Encryption Standard* (AES) adalah algoritma kunci yang mengenkripsi dan mendekripsi data menggunakan panjang kunci umum 128, 192, dan 256 bit [24]. Proses algoritma enkripsi AES terdiri dari 4 fungsi transformasi yaitu; *SubBytes*, *ShiftRows*, *MixColumns* dan *AddRoundKey* [25]. Untuk lebih detailnya, dapat dilihat pada Tabel 2.2 [26].

Tabel 2. 2 Tabel Perbandingan Enkripsi AES Berdasarkan Ukuran Bit

No	Versi AES	*Key Length*	*Block Size*	*Number of Rounds*
1.	AES-128	4	4	10
2.	AES-192	6	4	12
3.	AES-256	8	4	14

Secara umum, Pada awal proses enkripsi, input yang telah disalin ke dalam state akan mengalami transformasi *byte AddRoundKey*. Setelah itu, status akan beralih melalui transisi *SubBytes*, *ShiftRows*, *MixColumns*, dan *AddRoundKey* dari *Number of Rounds*. Proses dalam algoritma AES ini disebut *round-function*. Pada perputaran terakhir sedikit berbeda dengan perputaran sebelumnya dimana *state* tidak mengalami transisi *MixColumns* [27].

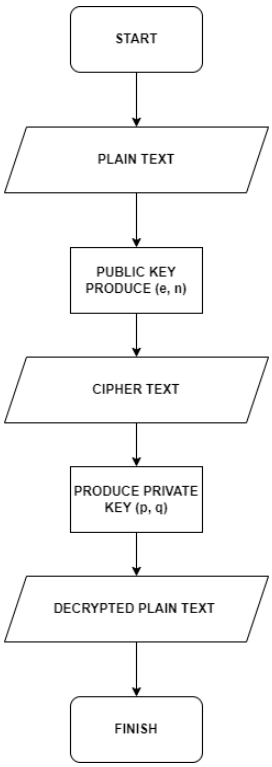


Gambar 2. 3 Diagram Skema Enkripsi AES 256

Pada penelitian ini akan digunakan enkripsi AES-256 dimana salah satu mode kerja AES adalah CBC, atau yang disebut *Cipher Block Chaining*. Pada algoritma block cipher seperti AES, *plaintext* atau pesan mentah yang masuk diproses dengan panjang yang tetap, yaitu n, namun jika ukuran data terlalu panjang, data dibagi menjadi blok-blok yang lebih kecil. Dalam CBC, urutan bit *plaintext* dibagi menjadi blok-blok bit dengan panjang yang sama. Mode CBC membutuhkan IV (vektor inisialisasi) untuk digabungkan dengan plaintext pertama, dan blok ciphertext sebelumnya menjadi IV dari blok berikutnya [28].

2.2.3 Algoritma Rivest Shamir Adleman (RSA)

Rivest Shamir Adleman atau lebih sering disebut RSA merupakan kependekan dari nama para pencipta algoritma satu ini yaitu Ron Rivest, Adi Sahmir, Leonard Adleman [29]. Terdapat proses utama dalam sistem algoritma kriptografi RSA, yaitu proses produksi kunci privat dan kunci publik, yang masing-masing digunakan dalam proses enkripsi dan dekripsi [30].



Gambar 2. 4 Alur Algoritma RSA

Terdapat beberapa unsur penting dalam algoritma RSA [20]:

1. p dan q yang merupakan bilangan prima (bersifat rahasia)
2. $n = p.q$ (bersifat tidak rahasia)
3. $_{\varphi}(n) = (p-1)(q-1)$ (bersifat rahasia)
4. e = *encryption key* (bersifat tidak rahasia)
5. d = *decryption key* (bersifat rahasia)
6. m = *plaintext* yang merupakan teks awal (bersifat rahasia)

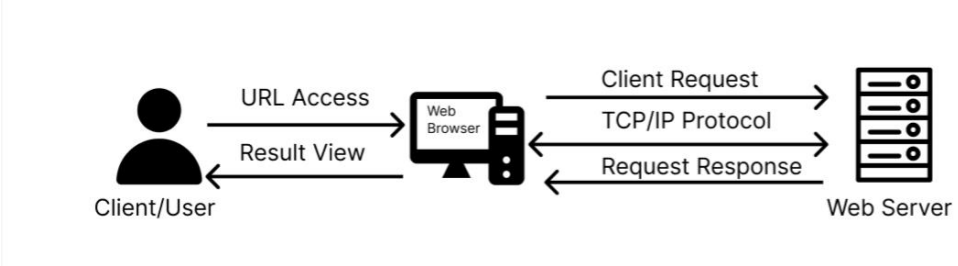
7. c = *chipertext* yang mana merupakan teks yang sudah dienkripsi (bersifat tidak rahasia)

2.2.4 *Web Server*

Web Server merupakan teknologi perangkat lunak yang menjadi penyedia layanan web [31]. *Web Server* bekerja menggunakan protokol HTTP dan HTTPS dan mengirimkan respon layanan kepada klien dalam sebuah halaman dengan format HTML [32]. Berkas, teks, video, gambar, dan file adalah bentuk halaman web yang dapat diterima oleh klien setelah melakukan *request* [33]. Dalam penerapannya, *Web Server* memiliki berbagai macam jenis seperti *Apache*, *Nginx*, *Sun Java System* dan masih banyak lagi.

2.2.5 Layanan Web

Layanan Web merupakan sebuah layanan yang tersimpan dalam *Web Server*. Layanan web atau *web service* merupakan teknologi software terdistribusi yang memiliki instrumen yang dapat dipasang pada *Web Server* dan diakses menggunakan protokol *HTTP* maupun *HTTPS* [34]. Cara kerjanya adalah *web server* akan memproses setiap *request* yang masuk serta menanggapi permintaan tersebut. Pengguna atau *client* akan menerima hasil *request* melalui platform *web browser* dimana fungsi dari web browser ini sendiri adalah menerima informasi dari web server untuk kemudian menampilkan hasilnya kepada pengguna atau *client*.



Gambar 2. 5 Cara Kerja Layanan Web

Web bekerja dengan menghubungkan antara *server* dan *client* pada suatu jaringan internet. Pengguna dapat memasukkan alamat web yang dituju. Dengan

menggunakan protokol TCP/IP permintaan dan pengiriman dikirimkan setelah dilakukan adanya pengecekan kesesuaian antara data pengiriman dan permintaan. Jika kesesuaian terpenuhi, data akan ditampilkan pada *web browser*.

2.2.6 Serangan *SQL Injection*

Injeksi SQL, atau yang lebih dikenal sebagai *SQL Injection*, adalah teknik serangan yang menargetkan server yang menggunakan kode SQL untuk melakukan memanipulasi dari sisi *database* [35]. *SQL Injection* sendiri bekerja dengan memanfaatkan celah yang ada pada sistem dari sisi *database* [36]. Serangan *SQL Injection* terhadap korban dapat menyebabkan kerugian diantaranya adalah *username dan password bypass*, pencurian data dan informasi, bahkan memodifikasi *database* target. Berbagai bentuk serangan *SQL Injection* meliputi *Tauntology-Based, Incorrect Queries, Stored Procedure Injection, Piggy Backed, Union-Based, Error-Based,* dan *Blind SQL Injection* [37].

2.2.7 *Web Hosting*

Hosting, atau biasa disebut dengan *web hosting*, merupakan layanan persewaan ruang penyimpanan data yang digunakan untuk menyimpan data website agar halaman website dapat diakses dari mana saja [38]. Data yang disimpan pada *web hosting* dapat berupa gambar, email, dokumen, video, dan situs web [39]. Umumnya, penyewaan penyimpanan pada web hosting juga disertakan penawaran paket dengan penyewaan domain. Domain sendiri adalah nama situs web yang diluncurkan di browser dan bersifat unik [40]. Pada praktiknya, domain akan berperan sebagai *Uniform Resource Locator* (URL) yang umumnya dikenal sebagai tautan, *link*, atau alamat web.

2.2.8 PHP

PHP adalah bahasa pemrograman untuk membuat situs web yang bersifat *server-side script*. PHP sendiri bersifat dinamis. PHP dapat berjalan di beberapa sistem operasi seperti Windows, Linux dan Mac OS [41]. PHP berperan sebagai pengolah data di sisi server dan membuat website menjadi lebih interaktif dan

dinamis [42]. Pada sebuah *website* dengan *database MySQL*, PHP dapat digunakan sebagai koneksi *database* dan situs web. PHP memiliki ciri khas dari sisi penulisan dengan tag “<?php” sebagai pembuka dan tag “?>” sebagai penutup [43].

2.2.9 *Openssl Library*

OpenSSL pada dasarnya adalah satu alat yang mencakup dua fungsi yakni sebagai perpustakaan kriptografi dan *toolkit* SSL [44]. Bahasa pemrograman PHP sendiri sudah mendukung *library openssl* yang dapat secara langsung dipanggil untuk melakukan enkripsi dan dekripsi. Pada penerapan lain, Openssl dapat digunakan via *command prompt* untuk membuat kunci publik, kunci privat maupun sertifikat untuk enkripsi asimetris.