BAB 2
TINJAUAN PUSTAKA

2.1. Landasan Perusahaan

Pada landasan perusahaan ini menjelaskan tentang perusahaan atau tempat melakukan penelitian. Pada penelitian ini perusahaan yang dijadikan tempat penelitian adalah Webhade Creative.

2.1.1. Sejarah Perusahaan

Webhade creative merupakan sub divisi dari PT. Anugerah Cipta Tarapti yang diresmikan menjadi perusahaan software house dan terfokus *pada website* dan *mobile apps development*. Webhade telah memiliki lebih dari 50 klien yang tersebar di seluruh Indonesia dan Singapura.

2.1.2. Visi dan Misi Webhade Creative

Visi webhade adalah menjadi perusahaan terbaik dan berdaya saing dalam memberikan jasa dan solusi yang berbasis pada inovasi dengan menawarkan kemudahan serta kepuasan pada pelanggan yang berkelanjutan.

Sedangkan misi webhade adalah sebagai berikut :

1. Memberdayakan dan menciptakan peluang bagi SDM yang berkompetensi.
2. Memberikan solusi dengan memprioritaskan Kepuasan Pelanggan (Customer Satisfaction).
3. Mengoptimalkan penggunaan teknologi dengan mengembangkan riset yg terpadu, berkesinambungan dan terarah untuk meningkatkan kompetensi
4. Membangun kesejahteraan bersama, dengan kemitraan strategis dan bersinergi dengan prinsip saling menguntungkan
5. Bermanfaat bagi pemilik perusahaan, karyawan, dan lingkungan sekitar.

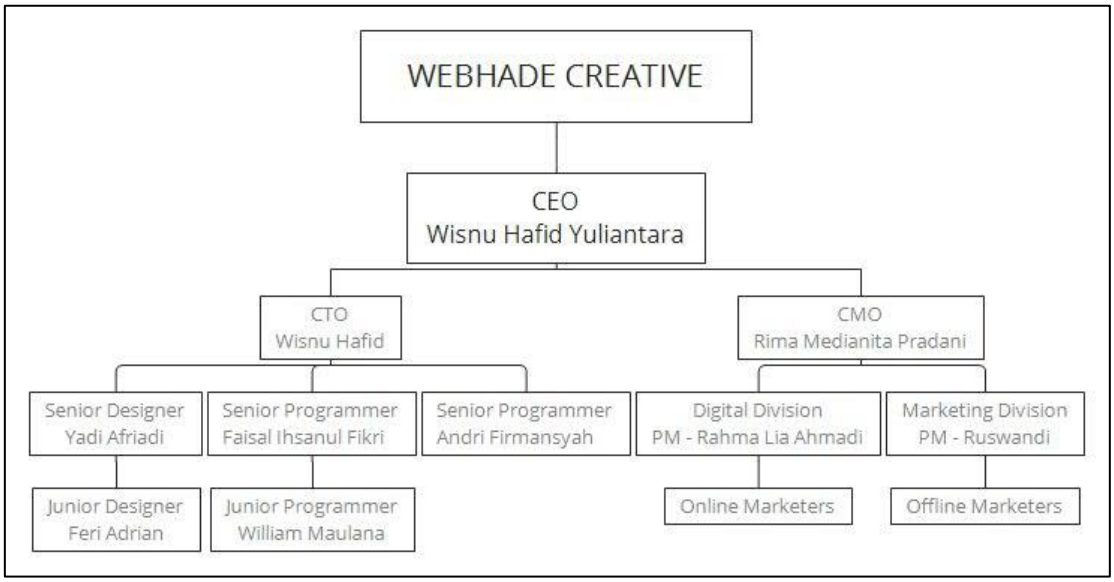
2.1.3. Webhade.com (Webhade Instant Builder)

Webhade Instant Builder adalah platform toko online yang dikembangkan untuk memenuhi kebutuhan UKM dalam menjalankan usahanya. Pengguna yang

mendaftar pada webhade instant builder akan mempunyai website sendiri yang digunakan sebagai media promosi usaha yang digeluti.

2.1.4. Struktur Organisasi

Berikut ini adalah struktur organisasi di Webhade Creative seperti pada Gambar 2.1.



Gambar 2. 1 Struktur Organisasi Webhade Cretive

2.2. Landasan Teori

Landasan teori ini menjelaskan hal – hal yang berkaitan dengan penelitian yang sedang dilakukan. Penelitian yang dilakukan kali ini ialah mengimplementasikan *advanced encryption standard* pada otorisasi *tenant database* di webhade creative.

2.2.1. Sejarah Algoritma

Kata algoritma berasal dari latinisasi nama seorang ahli matematika dari Uzbekistan yaitu Al Khawārizmi (hidup sekitar abad ke-9), sebagaimana tercantum pada terjemahan karyanya dalam bahasa latin dari abad ke-12 "Algorithmi de numero Indorum". Pada awalnya kata algorisma adalah istilah yang merujuk kepada aturan - aturan aritmetis untuk menyelesaikan persoalan dengan

menggunakan bilangan numerik arab (sebenarnya dari India). Pada abad ke-18, istilah ini berkembang menjadi algoritma, yang mencakup semua prosedur atau urutan langkah yang jelas dan diperlukan untuk menyelesaikan suatu permasalahan [15].

2.2.2. Definisi Algoritma

Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis. Kata Logis merupakan kata kunci dalam Algoritma. Langkah-langkah dalam Algoritma harus logis dan harus dapat ditentukan bernilai salah atau benar [16].

Dalam matematika dan komputasi, algoritma atau algoritme merupakan kumpulan perintah untuk menyelesaikan suatu masalah. Perintah-perintah ini dapat diterjemahkan secara bertahap dari awal hingga akhir. Masalah tersebut dapat berupa apa saja, dengan catatan untuk setiap masalah, ada kriteria kondisi awal yang harus dipenuhi sebelum menjalankan algoritma. Algoritma akan dapat selalu berakhir untuk semua kondisi awal yang memenuhi kriteria, dalam hal ini berbeda dengan heuristik. Algoritma sering mempunyai langkah pengulangan (iterasi) atau memerlukan keputusan (logika Boolean dan perbandingan) sampai tugasnya selesai. Programmer komputer akan lebih mudah menuangkan prosedur komputasinya atau urutan langkah proses dengan terlebih dahulu membuat gambaran (diagram alur) diatas kertas [16].

Algoritma dapat dituliskan dengan menggunakan banyak macam notasi, contohnya bahasa natural atau bahasa sehari-hari, pseudocode, diagram alir, bahasa pemrograman dan control tables. Penggunaan bahasa sehari-hari untuk menyatakan suatu urutan algoritma akan menimbulkan banyak keambiguan terhadap pernyataan yang dibuat, oleh karenanya pseudocode, diagram alir dan control tables menjadi lebih umum digunakan untuk menuliskan algoritma karena dapat menghindari keambiguan. Pertimbangan dalam pemilihan algoritma adalah algoritma haruslah benar. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. Tidak peduli serumit apapun algoritma, jika

memberikan keluaran yang salah, pastilah algoritma tersebut bukanlah algoritma yang baik [16].

Faktanya, setiap orang dapat membuat algoritma yang berbeda untuk menyelesaikan suatu permasalahan, namun tetap mengacu pada hasil atau keluaran yang sama.

2.2.3. Kriptografi

Kriptografi adalah salah satu metode yang digunakan untuk mengamankan data. Enkripsi akan mengubah data yang sebelumnya data yang mudah dibaca (*plaintext*) menjadi data yang sulit dibaca (*ciphertext*). Meskipun data berhasil di dapatkan oleh peretas, data tidak akan bisa dibaca karena yang didapatkan oleh peretas adalah data berupa cipheretxt. Dalam kriptografi terdiri dari dua hal, enkripsi (encryption) yaitu proses merubah data asli (plain text) menjadi data samaran (chiper text) dan dekripsi (decryption) yaitu proses pengembalian chiper text menjadi plain text kembali [6][7][8].

Untuk menjaga pesan, maka pesan tersebut dapat diubah menjadi suatu kode yang tidak dapat dimengerti oleh pihak lain. Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu [17]:

1. Enkripsi

   Merupakan hal yang sangat penting dalam kriptografi, merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut *plain text*, yang diubah menjadi kode-kode yang tidak dimengerti. Enkripsi bisa diartikan dengan *cipher* atau kode. Sama halnya dengan jika tidak mengerti akan sebuahkata maka yang dilakukan adalah dengan melihatnya di dalam kamus atau daftar istilah. Beda halnya dengan enkripsi, untuk mengubah teks asli ke bentuk teks-kode digunakan algoritma yang dapat mengkodekan data yang diinginkan.

2. Dekripsi

   Merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (*plaintext*), disebut dengan dekripsi pesan. Algoritma yang

digunakan untuk dekripsi tentu berbeda atau kebalikan dengan algoritma yang digunakan untuk enkripsi.

3. Kunci

   Kunci yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).

4. *Plaintext*

   Plaintext merupakan pesan yang ditulis atau diketik yang memiliki makna. *Plaintext* inilah yang diproses menggunakan algoritma kriptografi untuk menjadi *ciphertext*.

5. *Ciphertext*

   Merupakan suatu pesan yang telah melalui proses enkripsi. Pesan yang ada pada *ciphertext* ini tidak bias dibaca karena berupa karakter-karakter yang tidak mempunyai makna (arti)

6. Kriptanalisis *(Cryptanalysis)*

   Kriptanalisis merupakan suatu analisis kode atau suatu ilmu untuk mendapatkan *plaintext* tanpa harus mengetahui kunci sah secara wajar. Jika suatu *ciphertext* berhasil diubah menjadi *plaintext* tanpa menggunakan kunci yang sah, proses tersebut dinamakan breaking code. Hal ini dilakukan oleh para kriptanalis. Analisis kode juga dapat menemukan kelemahan dari suatu algoritma kriptografi dan akhirnya dapat menemukan kunci atau *plaintext* dari *ciphertext* yang dienkripsi dengan algoritma tertentu.

Menurut Bruce Schneier pada bukunya Applied Cryptography, Kriptografi adalah ilmu dan sekaligus seni untuk menjaga kerahasiaan berita. Selain pengertian tersebut, kriptografi juga sering diartikan sebagai ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data [18].

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu :

1. Kerahasiaan

   Layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka/mengupas informasi yang telah disandi.

2. Integritas data

   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. Autentikasi

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

4. Non Repuidasi

   Usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengirimkan/membuat.

2.2.4. Algoritma Kriptografi

Algoritma sandi (algoritma kriptografi) adalah algoritma yang berfungsi untuk melakukan tujuan kriptografis. Menurut Shannon, Algoritma kriptografi harus memiliki kekuatan untuk melakukan [19] :

1. Konfusi / pembingungan (confusion), dari teks terang sehingga sulit untuk direkonstruksikan secara langsung tanpa menggunakan algoritma dekripsinya.

2. Difusi / peleburan (difusion), dari teks terang menjadi ciphertext sehingga karakteristik dari teks terang tersebut hilang.

Pada implementasinya sebuah algoritma kriptografi harus memperhatikan kualitas layanan / Quality of Service atau QoS dari keseluruhan sistem dimana dia diimplementasikan. Algoritma kriptografi yang handal adalah algoritma kriptografi yang kekuatannya terletak pada kunci dan distribusi kunci, bukan pada kerahasiaan algoritma itu sendiri. Teknik dan metode untuk menguji kehandalan algoritma kriptografi adalah kriptanalisa [20].

Dasar matematis yang mendasari proses enkripsi dan dekripsi adalah relasi antara dua himpunan yaitu yang berisi elemen teks terang / plaintext dan yang berisi elemen teks sandi / ciphertext. Enkripsi dan dekripsi merupakan fungsi transformasi antara himpunan-himpunan tersebut. Apabila elemen-elemen teks terang dinotasikan dengan P, elemen-elemen teks sandi dinotasikan dengan C, sedang untuk proses enkripsi dinotasikan dengan E, dekripsi dengan notasi D [21].

Enkripsi : $E(P) = C$

Dekripsi : $D(C) = P$ atau $D(E(P)) = P$

Secara umum berdasarkan kesamaan kuncinya, algoritma kriptografi dibedakan menjadi :

1. Kunci-simetris / *symetric-key*, kunci yang digunakan untuk melakukan enkripsi sama dengan kunci yang digunakan untuk dekripsi
2. Kunci-asimetris / *asymetric-key* , kunci enkripsi dan kunci dekripsi tidak sama

Berdasarkan kerahasiaan kuncinya, algoritma kriptografi dibedakan menjadi :

1. Algoritma kriptografi kunci rahasia secret-key
2. Algoritma kriptografi kunci publik publik-key

2.2.4.1. Algoritma Simetris

Algoritma ini sering disebut dengan algoritma klasik karena memakai kunci yang sama untuk kegiatan enkripsi dan dekripsi. Algoritma ini sudah ada sejak lebih

dari 4000 tahun yang lalu. Bila mengirimkan pesan dengan menggunakan algoritma ini, si penerima pesan harus diberitahu kunci dari pesan tersebut agar bisa mendekripsikan pesan yang dikirim. Keamanan dari pesan yang menggunakan algoritma ini tergantung pada kunci. Jika kunci tersebut diketahui oleh orang lain maka orang tersebut akan dapat melakukan enkripsi dan dekripsi terhadap pesan [17]. Algoritma simetris terbagi menjadi dua, diantaranya adalah block cipher dan stream cipher.

Algoritma *block cipher* merupakan algoritma yang masukan dan keluarannya berupa satu blok dan setiap blok terdiri dari beberapa bit (misalnya 1 blok terdiri dari 64 bit atau 128 bit). *Blok cipher* mempunyai banyak aplikasi. Aplikasi tersebut digunakan untuk memberikan layanan confidentiality (kerahasiaan) integritas data atau authentication (pengesahan pemakai) dan juga bisa memberikan layanan *keystream generator* untuk *stream cipher* [17].

Stream cipher (aliran cipher) merupakan suatu *cipher* yang berasal dari hasil XOR. Setiap bit *plaintext* dengan setiap bit kunci. Kunci merupakan kunci utama (kunci induk) yang digunakan untuk membangkitkan kunci acak semu yang dibangkitkan dengan *Pseudo-Random Sequence Generator* yang merupakan suatu nilai yang nampak seperti diacak, tetapi sesungguhnya nilai tersebut merupakan suatu urutan. Secara khusus urutan dari nilai yang dihasilkan oleh *RNG (random number generator), computational mekanisme deterministic* atau FSM (*Finite State Machine*) merupakan kebalikan dari *really random* [17].

2.2.4.2. Algoritma Asimetris

Algoritma asimetris sering juga disebut dengan algoritma kunci publik, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsi berbeda. Pada algoritma asimetris kunci terbagi menjadi dua bagian yaitu [17]:

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

Kunci-kunci tersebut berhubungan satu sama lain. Dengan kunci publik orang dapat mendekripsi pesan tetapi tidak bisa mendekripsinya. Hanya orang yang memiliki kunci rahasia yang dapat mendekripsi pesan tersebut. Algoritma asimetris bias mengirimkan pesan dengan lebih aman daripada algoritma simetris. Contoh algoritma yang menggunakan kunci asimetris adalah RSA, Digital Signature Algorihtm (DSA) dan Elliptic Curve Cryptography (EGC).

2.2.5. Advanced Encryption Standard (AES)

AES merupakan algoritma lanjutan dari DES sebagai algoritma standar internasional untuk menggantikan DES yang telah dipecahkan pada tahun 2001. AES menyerupai DES yang merupakan algoritma *block cipher* dimana algoritma ini akan membagi teks menjadi blok yang telah ditetapkan panjang bit nya. Jika pada DES panjang kunci hanya maksimal 56bit, sedangkan pada AES panjang kunci terbagi menjadi 3 jenis yaitu 128bit dengan 10 perulangan algoritma, 192bit dengan 12 kali perulangan algoritma dan 256bit dengan 14 kali perulangan algoritma [13].

Tabel 2. 1 *Box Sub Bytes*

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

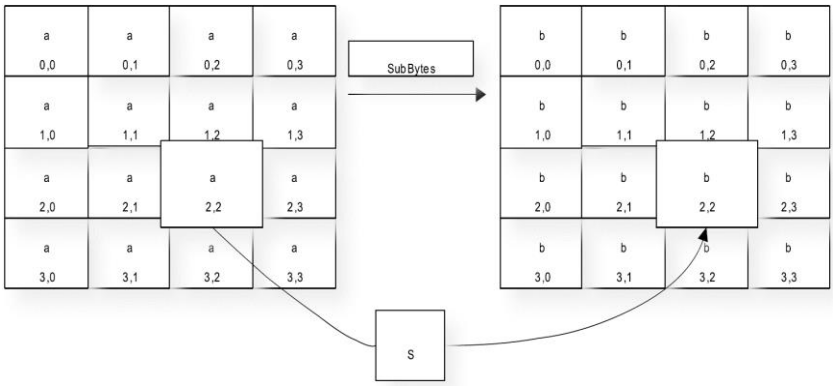
Menurut Rinaldi Munir, algoritma AES menggunakan substitusi dan permutasi, dan sejumlah putaran (*chiper* berulang), dimana setiap putaran menggunakan kunci yang berbeda (kunci setiap putaran disebut *round key*). AES

menetapkan panjang kunci 128, 192, dan 256 bit. Karena itu, maka dikenal AES-128, AES-192, dan AES- 256.

Garis besar Algoritma AES yang beroprasi pada blok 128- bit, dengan kunci 128-bit dapat dilihat pada gambar ,adalah sebagai berikut (di luar proses pembangkitan round key).

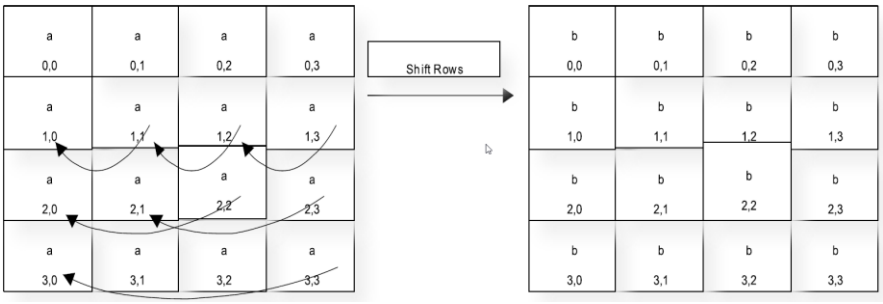
1. *AddroundKey*: Melakukan XOR antara *state* awal (*plaintext*) dengan *chiperkey*. Tahapan ini sering disebut juga *initial round*.
2. *Round* : Putaran sebanyak Nr-1 Putaran sebanyak Nr-1 kali. Proses yang dilakukan pada setiap putaran adalah:

   a. *SubBytes* : Putaran *byte* dengan menggunakan tabel substitusi.
      Proses SubBytes memetakan setiap *byte* dari array *State* dengan menggunakan tabel substitusi *S-Box*. Tidak seperti Des *S-box* berbeda pada setiap putaran, AES hanya mempunyai satu buah *S-Box*.



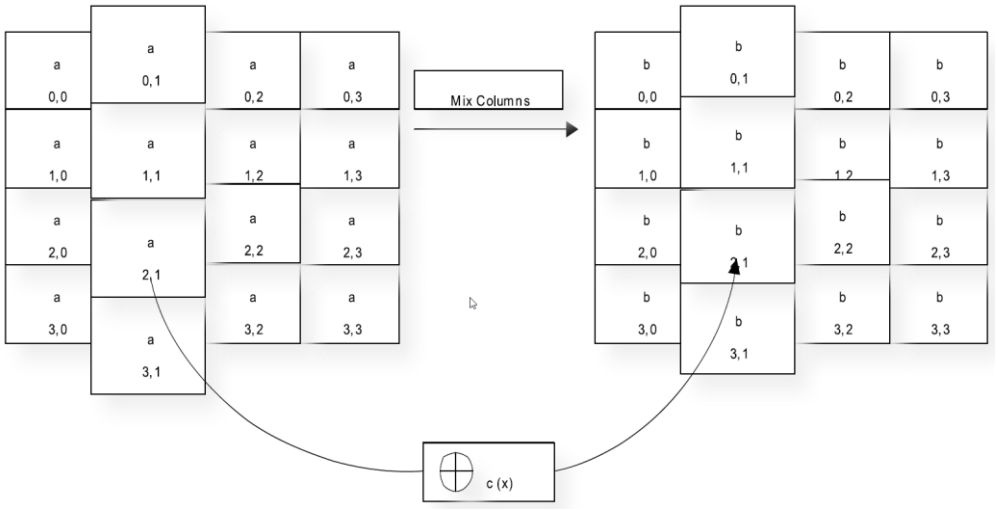
Gambar 2. 2 Transformasi *SubBytes*

   b. *ShiftRows*: Pergeseran barisbaris array *state* secara *wrapping*.
      Melakukan pergeseran secara *wrapping* (siklik) pada baris terakhir dari array *state*. Jumlah pergeseran bergantung pada nilai baris (r). Baris r=1 digeser sejauh 1 byte, baris r=2 digeser sejauh 2 byte Dan baris r=3 digeser sejauh 3 byte. Baris r=0 tidak digeser.



Gambar 2. 3 Transformasi *ShiftRows*

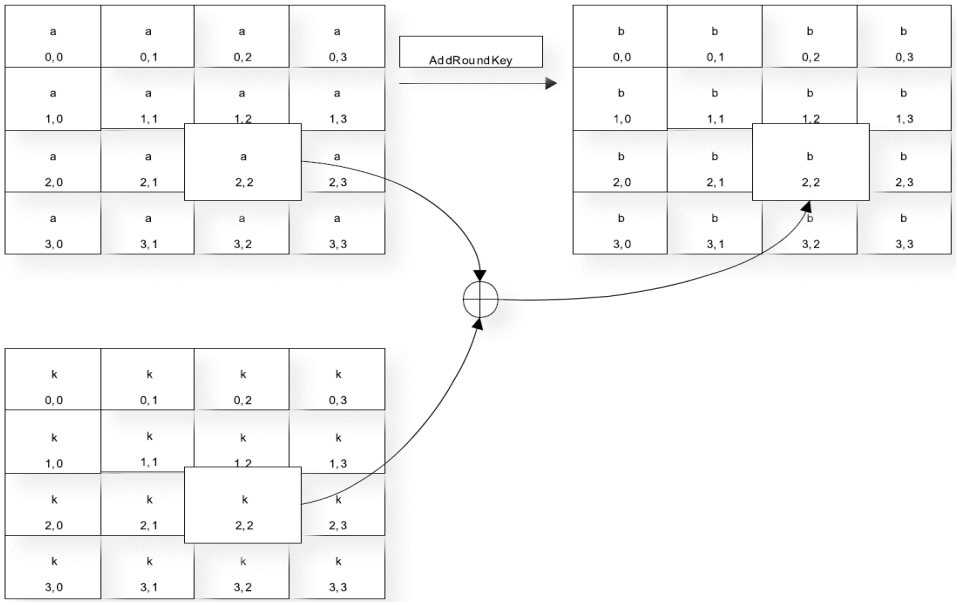
c. *MixColumn* : Mengacak data di masing-masing kolom *array state*.



Gambar 2. 4 Transformasi *MixColumn*

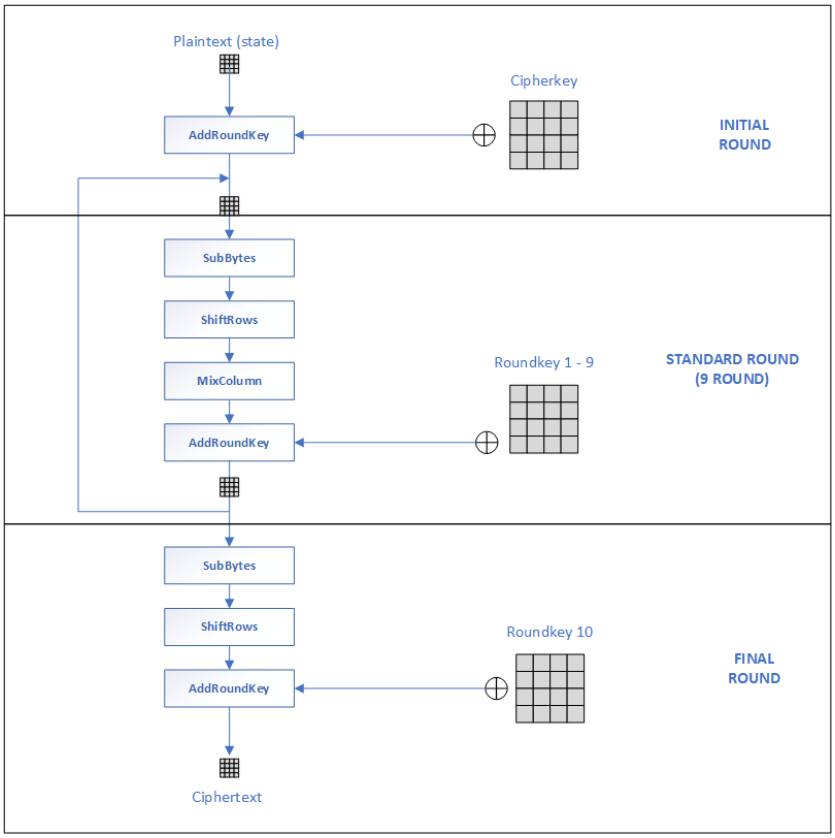
d. *AddRoundKey*: Melakukan XOR antara *state* sekarang *roundKey* seperti pada gambar 2.5.

3. *Final Round* : Proses untuk putaran terakhir.
   a. SubBytes
   b. ShiftRows
   c. AddRoundKey



Gambar 2. 5 Transformasi *AddRoundKey*

Untuk lebih jelas mengenai alur proses enkripsi pada aes 128-bit dapat dilihat pada gambar 2.6.



Gambar 2. 6 Proses Enkripsi Advanced Encryption Standard

Transformasi SubBytes, ShiftRows, MixColumn dan AddRoundKey dilakukan sebanyak 9 kali puraran. Pada terakhir hanya dilakukan SubBytes, ShiftRows dan AddRoundKey saja.

2.2.6. Base64

Base64 sejatinya bukan enkripsi, namun hanyalah sebuah standar penyandian (encoding). *Encoding* merupakan sebuah metode yang bertujuan untuk merubah bentuk atau format data. Beberapa contoh dari *encoding* adalah base64. Base64 berawal dari surat elektronik (*email*). Pada waktu itu, email dikirim dengan protokol SMTP *(Simple Mail Transfer Protocol)* ke *mail server* pengirim, lalu dikirim ke *mailbox* penerima di *mail server* tujuan. "Protokol" adalah tata cara mesin (komputer) saling berkomunikasi via jaringan. Supaya *email* bisa sampai ke penerima, ia harus mengunduhnya terlebih dahulu. Proses mengunduh email menggunakan protokol POP *(Post Office Protocol).* Saat ini POP sudah mencapai versi 3 sehingga disebut POP3. Alternatif yang lebih baik dari POP adalah IMAP *(Internet Mail Access Protocol),* yang saat ini sudah mencapai versi 4. POP dan SMTP memiliki persamaan dalam hal terminasi pesan. Keduanya menggunakan deretan karakter [CR LF . CR LF] *(carriage return – line feed – tanda titik – carriage return – line feed)* sebagai akhir dari pesan. Apa yang terjadi bila di dalam file biner terdapat *byte-byte* berikut: [0D 0A 2E 0D 0A]? Nilai rangkaian *byte* tadi adalah kode ASCII dari [CR LF . CR LF] sehingga server akan menganggap pesan yang dikirim berhenti sampai di sana. File yang dilampirkan oleh pengirim akan putus di tengah. Sehingga untuk mengatasi hal tersebut dibuat Base64 [22].

Tabel 2. 2 Indeks Base64 [5]

Indeks (data 6 bit)	**Karakter *Encoding* Base64**	**Indeks (data 6 bit)**	**Karakter *Encoding* Base64**	**Indeks (data 6 bit)**	**Karakter *Encoding* Base64**	**Indeks (data 6 bit)**	**Karakter *Encoding* Base64**
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3

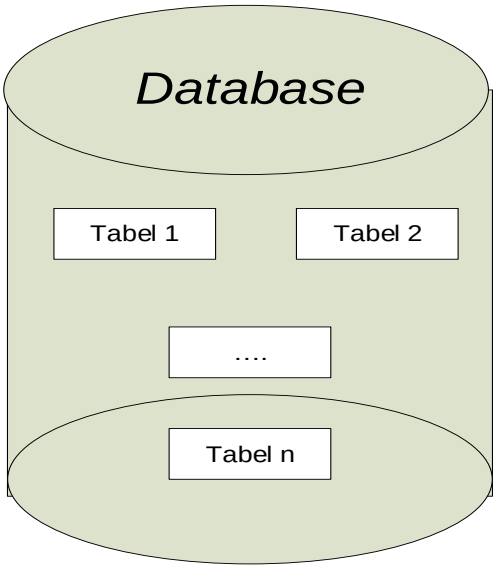
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/
						pad	=

Base64 adalah metode untuk melakukan penyandian (encoding) terhadap data binary menjadi format 6-bit character. Pada algoritma ini, rangkaian bit-bit palainteks dibagi menjadi blok-blok bit dengan panjang yang sama, biasanya 64 bit yang direpresentasikan dengan karakter ASCII. Base64 menggunakan karakter A – Z, a – z dan 0 – 9 untuk 62 nilai pertama, sedangkan 2 nilai terakhir digunakan symbol (+ dan /)sehingga totalnya 64 [23].

2.2.7. Database

Data adalah *value* yang turut merepresentasikan deskripsi dari suatu objek atau kejadian. Satuan terkecil data adalah bit. Bit adalah *binary digit* adalah basis angka yang terdiri dari angka O dan 1. Satuan ini menggmbarkan seberapa banyak bit (angka O dan 1) yang dapat mengalir dari satu tempat ke tempat yang lain dalam setiap detiknya melahui suatu media [24]. Semua data komputer disimpan dalam angka-angka biner. Hanya 2 nilai berbeda yang bisa dinyatakan satu bit, entah nilai 0 atau nilai 1.

Sedangkan *database* adalah kumpulan dari item data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu, tersimpan di hardware komputer dan dengan *software* untuk melakukan manipulasi untuk kegunaan tertentu [25][26]. *Database* biasanya disimpan di server yang bertugas menyediakan berbagai jenis kebutuhan klien [27]. Performa *database* ditentukan oleh kualitas server yang digunakan [28].



Gambar 2. 7 Konsep Umum Database [29]

Sebuah *database* relasional tersusun atas sejumlah tabel. Sebagai contoh, *database* akademis mencakup tabel-tabel seperti dosen, mahasiswa, KRS, nilai dan lain-lain. Database tentang bintang film bisa mencakup info pribadi (nama, jenis kelamin, tanggal lahir, dan sebagainya) dan film-film yang pernah dibintangi [29].

Tenant database pada sistem adalah sistem atau aplikasi yang menggunakan lebih dari satu *database*. penggunaan database dengan jumlah lebih dari satu dimaksudkan untuk membagi atau segmentasi data, berdasarkan klasifikasi tertentu atau untuk meningkatkan performa suatu aplikasi. Konsep tenant disini juga diartikan sebagai arsitektur perangkat lunak yang melayani banyak pengguna. Dengan konsep ini, sebuah perangkat lunak dirancang untuk memiliki partisi data yang berbeda dan dapat dikonfigurasi [30].

2.2.7.1. Komponen Penyusun Database

Dalam suatu *database*, terdiri dari beberapa komponen yang menyusunnya yaitu [31]:

1. *Entity*

Entity disebut juga dengan Tabel. *Entity* adalah orang, tempat, kejadian aatau konsep yang informasinya direkam.

2. *Atribute*

Setiap *entity* mempunyai *atribute* atau sebutan untuk mewakili suatu *entity*. *Atribute* juga disebut sebagai *data elemen, data field, data item.*

3. *Data Value* (Nilai atau Isi Data)

Data value adalah data aktual atau informasi yang disimpan pada tiap data elemen atau atribute.

4. *Record/Tuple*

Record/tuple merupakan kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu *entity* secara lengkap. Satu *record* mewakili satu data atau informasi.

5. Relasi/Hubungan

Pada model *database* relasional, kaitan atau asosiasi antara dua buah tabel disebut hubungan *(relationship).* Hubungan dapat berupa:

1) 1-1, yakni satu data pada suatu tabel berpasangan dengan hanya satu data pada tabel lain
2) 1-M, yani satu data pada suatu tabel berpasangan dengan banyak data pada tabel lain.

2.2.7.2. Fungsi Database

Penyusunan suatu database digunakan untuk mengatasi masalah-masalah pada penyusunan data yaitu [31]:

1. Redudansi dan inkonsistensi data dalam beberapa file.

Redudansi merupakan kondisi di mana terdapat data yang sama tersimpan di dalam beberapa tempat (duplikasi data). Hal ini mengakibatkan pemborosan ruang penyimpanan dan juga biaya untuk mengakses jadi lebih tinggi. Penyimpanan data yang sama berulang-ulang di beberapa file dapat mengakibatkan juga inkonsistensi (tidak konsisten).

2. Kesulitan pengaksesan data

Pada suatu saat dibutuhkan suatu proses untuk menampilkan data tertentu namun belum tersedia program untuk mengeksekusinya. Dengan adanya DBMS,

data mampu diambil secara langsung dengan bahasa yang familiar dan mudah digunakan (user friendly)

3. Isolasi data untuk standarisasi

Jika data tersebar dalam beberapa file dalam bentuk format yang tidak sama, maka ini menyulitkan dalam menyusun program aplikasi terutama proses mengambil dan menyimpan data. Maka haruslah data dalam satu database dibuat satu format sehingga mudah dibuat program aplikasinya.

4. Multiple user (banyak pemakai)

Dalam rangka mempercepat semua daya guna sistem dan mendapat responsi waktu yang cepat, beberapa sistem mengizinkan banyak pemakai untuk meng"update" data secara simultan. Salah satu alasan mengapa database dibangun karena nantinya data tersebut digunakan oleh banyak orang dalam waktu yang bervariasi.

5. Masalah keamanan (security)

Tidak setiap pemakai sistem database diperbolehkan untuk mengakses semua data. Misalkan data mengenai gaji pegawai hanya boleh dibuka oleh bagian keuangan dan personalia, tidak diperkenankan bagian gudang membaca dan mengubahnya.

6. Masalah integrasi (kesatuan)

Dengan adanya database, antara data satu sama lain yang berhubungan dapat dikaitkan. Secara teknis pengaitan tersebut dilakukan menggunakan field kunci.

7. Masalah data independence (kebebasan data)

Paket bahasa yang diciptakan oleh DBMS, perubahan pada struktur file/tabel, setiap kali kita hendak melihat data cukup dengan utility list, menambah data dengan Append (misal untuk DBMS Clipper atau Foxpro), merubah struktur tabel dengan Design Table, melakukan penelurusan data dengan query (misal untuk Access, Sql Server, MySql atau Oracle). Ini berarti perintah-perintah dalam paket DBMS bebas terhadap database. Apapun perubahan dalam database, semua perintah akan mengalami kestabilan tanpa perlu ada yang diubah.

2.2.7.3. Entity Relational Diagram (ERD)

Model yang menggambarkan hubungan antara data yang disimpan dikenal sebagai diagram relasional entitas (ERD). ERD adalah yang paling banyak digunakan dari semua alat pemodelan untuk mencerminkan data yang disimpan sistem.

Entitas dalam basis data adalah objek yang memiliki data atau atribut. Dalam basis data aplikasi ritel, pelanggan, produk, dan pemasok mungkin merupakan entitas. Entitas dapat menggolongkan sejumlah atribut, atribut produk mungkin warna, ukuran, dan harga, atribut pelanggan mungkin termasuk nama, alamat, dan peringkat kredit.

ERD dapat digunakan untuk keperluan pemodelan data, bahwa suatu entitas dapat dengan mudah didefinisikan sebagai file logis yang akan mencerminkan elemen data atau bidang yang merupakan komponen dari setiap kejadian atau catatan dalam entitas. Oleh karena itu, menunjukkan bagaimana banyak entitas akan berinteraksi satu sama lain. Kami menyebut interaksi ini sebagai "hubungan." Dalam kebanyakan kasus, hubungan antara dua entitas atau lebih hanya dapat ada jika mereka memiliki setidaknya satu elemen data umum di antara mereka. Terkadang kita harus memaksakan suatu hubungan dengan menempatkan elemen data umum antara entitas, hanya agar mereka dapat berkomunikasi satu sama lain [32].

2.2.8. Sniffing

Sniffing erat kaitannya dengan jaringan komputer atau internet. Internet adalah jaringan komputer berskala luas dan dapat diakses dari mana saja [33] [34]. Maka dari itu sangat dimungkinkan data yang ada pada proses transfer data dapat dilihat dengan teknik tertentu, salah satunya dengan teknik sniffing. *Sniffing* merupakan proses pengendusan paket data pada sistem jaringan komputer, yang diantaranya dapat memonitor dan menangkap semua lalu lintas jaringan yang lewat tanpa peduli kepada siapa paket itu di kirimkan. Contoh dampak negatif *sniffing*, seseorang dapat melihat paket data informasi seperti *username* dan *password* yang lewat pada jaringan komputer. Contoh dampak positif *sniffing*. Seorang admin

dapat menganalisa paket-paket data yang lewat pada jaringan untuk keperluan optimasi jaringan, seperti dengan melakukan penganalisaan paket data, dapat diketahui dapat membahayakan performa jaringan atau tidak, dan dapat mengetahui adanya penyusup atau tidak [35].

2.2.8.1. Wireshark

Wireshark adalah *tool* yang ditujukan untuk penganalisaan paket data jaringan. Wireshark disebut juga *Network packet analyzer* yang berfungsi menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi dipaket tersebut sedetail mungkin. Sebenarnya *network packet analyzer* sebagai alat untuk memeriksa apa yang sebenarnya terjadi di dalam jaringan baik kabel maupun *wireless*. Dengan adanya wireshark ini semua sangat dimudahkan dalam hal memonitoring dan menganalisa paket yang lewat di jaringan [35].

Ada beberapa contoh penggunaan Wireshark :

1. Admin sebuah jaringan menggunakan untuk *troubleshooting* masalah di jaringan.
2. Admin menggunakan Wireshark untuk mengamankan jaringannya.

Beberapa fitur kelebihan Wireshark, diantaranya :

1. Berjalan pada sistem operasi Linux dan Windows.
2. Menangkap paket (*Capturing Packet*) langsung dari *network interface*.
3. Mampu menampilkan hasil tangkapan dengan detail.
4. Dapat melakukan penyaringan paket
5. Hasil tangkapan dapat di simpan, di *import* dan di *export*.

2.2.8.2. Postman

Postman ini merupakan *tool* bagi para *developer* yang berkutat pada pembuatan *API*, fungsi utama postman ini adalah sebagai *GUI API Caller* namun sekarang postman juga menyediakan fitur lain, yaitu *Sharing Collection API for Documentation (free), Testing API (free), Realtime Collaboration Team*

(paid), Monitoring API (paid), Integration (paid). Postman digunakan untuk unit percobaan dalam segi koneksi pada *backend* aplikasi tersebut, aplikasi ini untuk mencoba hasil respon yang diberikan oleh *backend* melalui *API*, hasil respon tersebut berbentuk JSON, dimana nanti akan dipakai pada aplikasi [36].

Postman sangat erat kaitannya dengan pengetesan request yang dapat dilakukan melalui protokil http. *Hypertext transfer protocol* (HTTP) merupakan protokol yang digunakan untuk jenis layanan *world wide web* (WWW) pada jaringan TCP/IP. Pengembangan HTTP dikoordinasi oleh konsorsium WWW dan IETF (*internet engineering task force*) dan dipublikasikan melalui kumpulan RFC (*request for comments*). RCF 2616 mendefinisikan HTTP/1.1 yang merupakan versi HTTP yang saat ini umum digunakan.[3] Sebuah HTTP client memulai request dengan membuat koneksi TCP (*Transmission Control Protocol*) menuju server (umumnya adalah port 80). Sedangkan HTTP server menunggu adanya pesan *request* pada port yang telah ditentukan. Setelah menerima *request* dari *client, server* kemudian mengirimkan status line antara lain "HTTP/1.1 200 OK". Setelah itu dilanjutkan dengan mengirimkan file yang diinginkan *client* beserta pesan kesalahan atau informasi lainnya. HTTP diidentifikasi menggunakan *uniform resource identifier* (URI) dengan format penulisan tertentu [37].

Protokol HTTP bersifat *request-response*, yaitu dalam protokol ini client menyampaikan pesan *request* ke web *server* , dan *web server* kemudian memberikan *response* yang sesuai dengan *request* tersebut. *Request* dan *response* dalam protokol HTTP disebut sebagai *request chain* dan *response chain*. Hubungan HTTP yang paling sederhana adalah terdiri atas hubungan langsung antara *user agent* dengan *server* [37]. Metode *request* yang dapat dilakukan pada http adalah GET, POST, PUT atau DELETE [38].

2.2.9. Keamanan Jaringan

Kemanan jaringan adalah kumpulan peranti yang dirancang untuk melindungi data ketika transmisi terhadap ancaman pengaksesan, pengubahan dan penghalangan oleh pihak yang tidak berwenang [39].

2.2.9.1. Layanan Kemanan Jaringan

Layanan-layanan keamanan jaringan didefinisikan berdasarkan kebutuhan yang harus disediakan untuk memenuhi permintaan terhadap keamanan jaringan. Berikut adalah pembahasan jenis-jenis layanan keamanan jaringan [17]:

1. Otentikasi (*Authentication*), bertujuan agar penerima informasi dapat memastikan keaslian pesan, bahwa pesan itu datang dari orang yang dimintai informasi. Dalam layanan otentifikasi terdapat dua layanan di dalamnya. Layanan pertama disebut dengan otentikasi entitas (*entity authentication*) yaitu layanan yang memberikan kepastian terhadap identitas sebuah entitas yang terlibat dalam komunikasi data. Sedangkan layanan kedua disebut dengan otentikasi terhadap keaslian data (*data origin authentication)* yaitu layanan yang memastikan sumber dari sebuah data.
2. Kendali Akses (*Acces Control*) adalah layanan keamanan jaringan yang menghalangi penggunaan tidak terotorisasi terhadap sumber daya. Pada aplikasi jaringan biasanya kebijakan kemampuan (baca, modifikasi, tulis dan eksekusi sebuah data/layanan sistem) ditentukan oleh jenis pengguna. Misalnya sebuah data rekam medik elektronik hanya dapat diakses oleh pasien dan paramedis yang terlibat. Kendali askses seringkali dilakukan dengan menggunakan kombinasi *user id* dan *password* ataupun dengan mekanisme lain.
3. Kerahasiaan Data *(Confidentiality)* merupakan layanan keamanan jaringan yang memproteksi data tertransmisi terhadap pengungkapan oleh pihak yang tidak berwenang. Atau bisa didefinisikan juga bahwa layanan kerahasiaan data merupakan usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. Kerahasiaan ini biasanya berhubungan dengan informasi yang diberikan ke pihak lain.
4. Keutuhan Data *(Integrity)* adalah layanan keamanan jaringan yang memastikan bahwa data yang diterima oleh penerima adalah benar-benar sama dengan data yang dikirim oleh pengirim. Ada dua jenis layanan keutuhan data yaitu keutuhan data dengan pemulihan dan tanpa pemulihan.

5. *Non-Repudiation* adalah layanan keamanan jaringan yang berhubungan dengan pengirim. Dalam hal ini bertujuan untuk menghindari penolakan atas penerimaan/pengiriman data yang telah terkirim. Sehingga pengirim tidak dapat mengelak bahwa dialah yang mengirim informasi.
6. Ketersediaan (*Availability*) adalah layanan sistem yang membuat sumber daya sistem tetap dapat diakses dan digunakan ketika ada permintaan dari pihak yang berwenang. Serangan terhadap sistem seperti *denial of services*. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi.

2.2.9.2. Ancaman Keamanan

Ancaman keamanan yang terjadi terhadap informasi adalah [17]:

1. *Interruption*: merupakan suatu ancaman terhadap *availability* informasi. Data yang ada dalam sistem komputer dirusak atau dihapus sehingga jika data atau informasi tersebut dibutuhkan maka pemiliknya akan mengalami kesulitan untuk mengaksesnya, bahkan mungkin informasi itu hilang.
2. *Interception*: merupakan ancaman terhadap kerahasiaan (*secrecy*). Informasi disadap sehingga orang yang tidak berhak dapat mengakses kokmputer di mana informasi tersebut disimpan.
3. *Modification*: merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil menyadap lalu-lintas informasi yang sedang dikirim dan kemudian mengubahnya sesuai keinginan orang tersebut.
4. *Fabrication*: merupakan ancaman terhadap integritas. Orang yang tidak berhak berhasil meniru atau memalsukan informasi sehingga orang yang menerima informasi tersebut menyangka bahwa informasi tersebut berasal dari orang yang dikehendaki oleh si penerima informasi.

2.2.9.3. Serangan Kemanan Jaringan

Sistem keamanan jaringan yang dioperasionalkan pada jaringan public rentan terhadap serangan oleh siapapun. Orang yang berusaha meruntuhkan

keamanan jaringan disebut *attacker* (penyerang). Secara umum serangan pada sistem keamanan jaringan dapat dikategorikan jadi 2 jenis serangan, yaitu [39]:

1. **Serangan Pasif**

   Serangan pasif tidak melakukan modifikasi data yang melintasi atau merusak sistem, penyerang hanyamembaca saja (read only). Karena tidak melakukan perubahan data dan menggangu sistem, serangan pasif lebih pada pencegahan daripada pendeteksian. Serangan yang dapat digolongkan sebagai serangan pasif diantaranya *snooping* dan *traffic analysis*.

2. **Serangan Aktif**

   serangan aktif dapat mengakibatkan perubahan data yang terkirim dan mengganggu jalannya sistem. Pada serangan aktif penyerang memperoleh kemampuan untuk mengubah data pada lalu lintas data selain kemampuan baca. Serangan yang dapat digolongkan sebagai serangan aktif diantaranya *masquerade, denial of service, replay dan modification*

2.2.10. Pengujian Sistem

Sistem adalah kumpulan dari komponen – komponen yang digabungkan menjadi satu untuk mencapai tujuan atau maksud tertentu [40] [41]. Sistem bisa dibuat untuk mengelola informasi agar mudah untuk mendapatkannya [42] [43]. Pengujian sistem adalah sebuah elemen sebuah topik yang memiliki cakupan luas dan sering dikaitkan dengan verifikasi dan validasi. Verifikasi mengacu pada sekumpulan aktivitas yang menjamin bahwa perangkat lunak mengimplementasikan dengan benar sebuah fungsi yang spesifik. Validasi mengacu pada sekumpulan aktivitas yang berbeda yang menjamin bahwa perangkat lunak yang dibangun dapat ditelusuri sesuai dengan kebutuhan pelanggan (customer) [44].

2.2.10.1.Black Box Testing

Yaitu menguji perangkat lunak yang dibangun dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui

apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

Untuk melakukan pengujian *black box* diperlukan kasus uji yang dibuat untuk mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai dengan spesifikasi yang dibutuhkan atau tidak. Kasus uji yang dibuat dalam pengujian *black box* memerlukan kasus benar dan kasus salah [44]. Sebagai contoh kasus proses terhubung ke tenant database, kasus uji:

a. Jika pengguna memasukkan *username* dan *password* benar

b. Jika pengguna memasukkan *username* dan *password* salah, misalnya *username* benar tapi *password* salah, atau sebaliknya, atau keduanya salah.

2.2.10.2. Pengujian Keamanan Enkripsi

Pengujian keamanan enkripsi dilakukan untuk mengetahui aman atau tidaknya enkripsi yang dikirimkan. Pengujian ini menggunakan perangkat lunak yaitu CrypTool dan Wireshark.

CrypTool merupakan perangkat lunak untuk melakukan kriptografi maupun kripanalisis yang dapat diunduh secara gratis dari portal CrypTool di www.cryptool.org. Pada awalnya *CrypTool* dirancang sebagai perangkat lunak internal dalam Deutsche Bank untuk menunjang pelatihan keamanan pada sisi teknologi informasi. *CrypTool* kemudian mengalami perkembangan menjadi proyek *open-source* yang mengakibatkan pengguna semakin banyak sehingga membantu pengembang meningkatkan kualitas *CrypTool* melalui masukan-masukan yang diberikan pengguna [45].

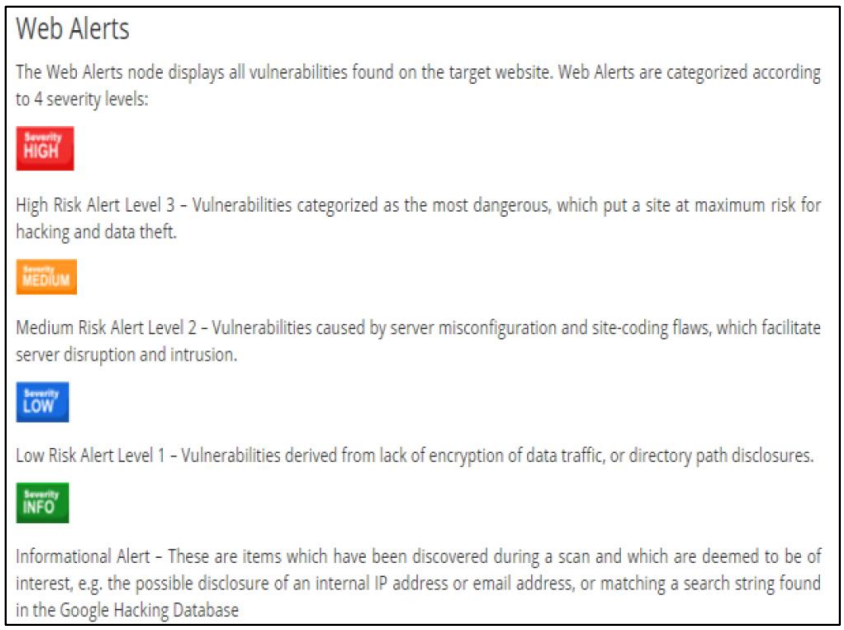
Wireshark adalah sebuah aplikasi yang dapat menangkap paket-paket jaringan dan berusaha untuk menampilkan semua informasi di paket tersebut sedetail mungkin. Aplikasi jenis ini merupakan. Dengan adanya wireshark ini semua sangat dimudahkan dalam hal memonitoring dan menganalisa paket yang lewat di jaringan [35].

2.2.10.3.Penetration Testing

Penetration testing atau uji penetrasi adalah suatu kegiatan dimana seseorang mencoba melakukan simulasi serangan yang bisa dilakukan terhadap jaringan organisasi atau perusahaan tertentu untuk menemukan kelemahan yang ada pada sistem tersebut. Orang yang melakukan kegiatan ini disebut penetration tester. Alat yang digunakan untuk melakukan penetrasi pada penelitian kali ini adalah acunetix web vulnerability scanner dan cryptool.

1. Acunetix Web Vulnerability Scanner

Acunetix Web Vulnerability Scanner adalah tool yang dapat digunakan untuk melakukan pemindaian terhadap suatu website untuk mengetahui vulnerability atau kelemahan yang ada pada website tersebut. Keamanan website mungkin aspek yang paling diabaikan saat ini. Padahal mengamankan perusahaan harus menjadi prioritas utama dalam setiap organisasi. Hacker berkonsentrasi mengusahakan upaya mereka pada aplikasi berbasis web (seperti shopping cart, forms, login page).



Gambar 2. 8 Level Tool Acunetix [46]

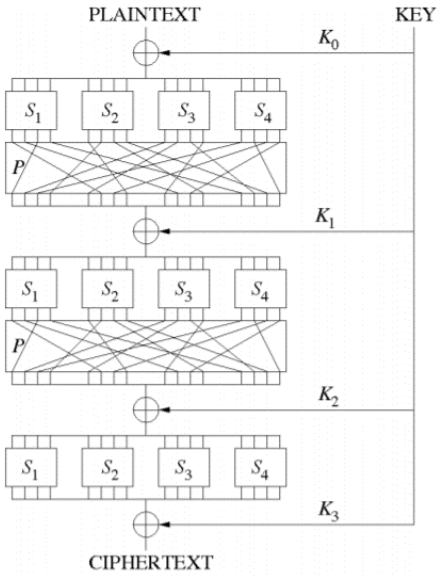
Web applications dapat diskses 24 Jam sehari, 7 hari seminggu dan bertugas untuk mengontrol data berharga karena web applications mempunyai akses langsung, sepertidatabase pelanggan. Web application sering dibuat namun kurang

dilakukan pengujian sehingga lebih mungkin mempmyai kerentanan yang kurang diperhatikan. Acunetix Web Vulnerability Scanner otomatls memeriksa web application terhadap SQL injection, XSS dan kerentanan web lainya. Tool Acnetix Web vmerability Scanner yang digunakan pada penelitian kali ini juga dapat menampilkan level dari hasil scanning [46].

Pada Gambar 2.8 dapat dilihat levels dari acunetix yang akan menjelaskan tingkat keamanan dari URL atau IP address yang di-scan.

2. CrypTool

CrypTool adalah platform e-learning open-source yang menyediakan pengguna dengan alat interaktif baik untuk mendapatkan wawasan ke dalam bidang kriptografi dan analisis kriptografi untuk amatir atau untuk memperdalam pemahaman mereka pada subjek untuk yang lebih berpengalaman.



Gambar 2. 9 Jaringan Substitusi Permutasi [47]

CrypTool menggunakan bahasa pemrograman C #, dan didasarkan pada Microsoft framework .NET dengan Windows Presentation Foundation WPF untuk membuat antarmuka pengguna dan lingkungan pengembangan Visual Studio. Dengan CrypTool, individu dapat menarik dan melepas plugin tunggal yang menerapkan berbagai algoritma kriptografi dan alat analisis kriptografi ke ruang kerja program dan menghubungkan mereka dengan komponen lain. Setiap plugin dilengkapi

dengan docking input dan output poin yang memungkinkan interaksi dan pertukaran data di antara mereka [47].

2.2.11. PHP

PHP adalah bahasa pemrograman script server-side yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP di kembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di http://www.php.net.

Pada awalnya PHP merupakan singkatan dari *Personal Home Page.* Sesuai dengan namanya, PHP digunakan untuk membuat website pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman web yang powerful dan tidak hanya digunakan untuk membuat halaman web sederhana, tetapi juga website populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dll.

Tahap perkembangan PHP dari awal versinya hingga sekarang adalah sebegai berikut :

1. Personal Home Page Tools (PHP Tools) version 1.0 rilis dan diumumkan pada 8 Juni 1995.
2. Personal Home Page/Forms Interpreter 2. Seiring dengan pengembangan dan penambahan fitur web pada saat itu, pada April 1996, Rasmus Lerdorf mengumumkan PHP/FI versi 2.0
3. PHP: Hypertext Preprocessor 3 rilis pada Juni 1998. Dengan dukungan dari banyak programmer lainnya, proyek PHP secara perlahan beralih dari proyek satu orang menjadi proyek massal yang lebih akrab kita kenal sebagai *open-source project.* PHP selanjutnya dikembangkan oleh The PHP Group yang merupakan kumpulan banyak programmer dari seluruh dunia. Perilisan PHP versi 3 juga ditandai dengan perubahan singkatan PHP yang sebelumnya Personal Home Page Tools, menjadi Hypertext Preprocessor

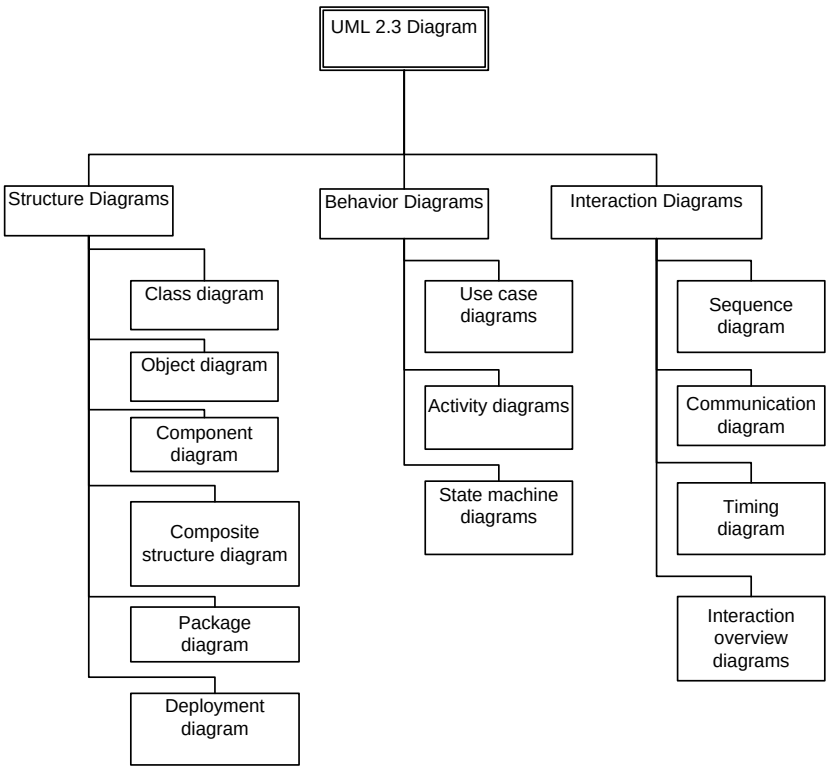
4. PHP: Hypertext Preprocessor 4 diluncurkan pada 22 May 2000. PHP versi 4 juga menyertakan fitur pemrograman objek/*Object Oriented Programming*, walaupun belum sempurna.
5. PHP: Hypertext Preprocessor 5
   Versi PHP terakhir hingga saat ini, yaitu PHP 5.x diluncurkan pada 13 Juli 2004. PHP 5 telah mendukung penuh pemrograman object dan peningkatan performa melalui Zend engine versi 2.
6. PHP: Hypertext Preprocessor 6
   Versi lanjutan dari PHP, yakni PHP 6.x sebenarnya telah lama dikembangkan, bahkan sejak tahun 2005. Fokus pengembangan PHP 6 terutama dalam mendukung Unicode, agar PHP bisa mendukung berbagai jenis karakter bahasa non-latin. Namun dikarenakan beberapa alasan seperti kurangnya programmer, dan performa yang tidak memuaskan, pengembangan PHP 6 dihentikan dan fitur yang ada dimasukkan kedalam PHP 5.
7. PHP: Hypertext Preprocessor 7. Pada tahun 2014, sebuah proyek lanjutan PHP mulai mengemuka, yakni PHP 7. PHP 7 berkembang dari proyek experimen yang dinamakan PHPNG (PHP Next Generation).

2.2.12. Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah salah satu standar bahasa yang banyak digunakana di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung [44].

1. Diagram UML

Pada UML 2.3 terdiri dari 13 macam diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar 2.8.



Gambar 2. 10 Diagram UML [44]

Kategori diagram UML:

1. *Structure Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari system yang dimodelkan
2. *Behavior Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan system atau rangkaian perubahan yang terjadi pada sebuah system.
3. *Interaction Diagrams* yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi system dnegan system lain maupun interaksi antar subsistem pada suatu system.

Berikut diagram yang akan digunakan dalam penelitian ini:

1. *Use case Diagram*

Use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah system dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat

dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian actor dan *use case* [44].

Aktor merupakan orang, proses atau system lain yang berinteraksi dengan system informasi yang akan dibuat di luar system itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang. *Use case* merupakan fungsionalitas yang disediakan system sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.

2. *Class Diagram*

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi [44].

3. Activity diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

Activity diagram juga banyak digunakan untuk mendefinisikan hal-hal berikut [44]:

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis system yang didefinisikan
2. Urutan atau pengelompokan tampilan dari sistem/*user interface* di mana ssetiap aktivitas dianggap memiliki sebuah rancangan antarmuka pilihan
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu yang ditampilkan pada perangkat lunak

4. *Sequence diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima

antar objek. Oleh karena itu untuck menggambar *sequence diagram* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki. Membuat *sequence diagram* juga dibutuhkan untuk melihat skenario yang ada pada *use case* [44].