BAB II
LANDASAN TEORI

1.1 Tinjauan Pustaka

Dalam penelitian ini menggunakan lima tinjauan pustaka yang akan menjadi pendukung pada penelitian ini, berikut adalah tinjauan studi yang membantu dalam penelitian ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Daftar Literatur

No Literatur	**Penulis**	**Tahun**	**Algoritma**	**Data**
Literatur 1	Iisma Nur Hanifah	2021	*Reed Solomon*	Presensi QR Code
Literatur 2	Muazharin Alfan, dkk	2021	RC4 *(Rivest Chiper)*	Enkripsi Data *QR-CODE* pada Aplikasi Presensi
Literatur 3	Nanda Dwi Wicaksono	2019	*Vigenere Chiper*	Implementasi *QR-CODE* dan *Vigenere Chiper* pada Sistem Manajemen Perkuliahan
Literatur 4	Prastyo Irwan, dkk	2021	Triple DES	Sistem Presensi Menggunakan *QR-CODE*
Literatur 5	Zakaria Adi Putra	2019	AES *(Advanced Encryption Standard)*	Presensi dengan Implementasi *QR-CODE*

1.1.1 Tinjauan Pustaka Literatur 1

Pada penelitian ini dengan menggunakan Algoritma *Reed Solomon* dibuat dengan tujuan pada saat encoding dan dengan menghitung tingkat kesalahan dengan *Reed Solomon* didalam *QR-CODE* pada aplikasi digunakan untuk mewujudkan *QR-CODE* online ke dalam sebuah sistem didalam aplikasi abesensi. (Iisma Nur Hanifah, 2019)

1.1.2 Tinjauan Pustaka Literatur 2

Perkembangan teknologi saat ini sangat pesat, sistem keamanan semakin banyak diimplementasikan dalam berbagai sistem dan aplikasi. Dalam penyampaian pesan tersembunyi contohnya sudah terdapat banyak metode atau algoritma yang semakin marak digunakan. Banyaknya proses pengiriman data terlebih melalui REST API sangat rentan untuk dicuri dan digunakan tidak semestinya. Maka dari itu banyak developer yang memilih untuk melakukan enkripsi pada data yang dikirim tersebut. *Rivest Ciphe*r 4 (RC4) merupakan jenis aliran kode yang berarti operasi enkripsinya dilakukan per karakter 1 *byte* untuk sekali operasi. Algoritma ini merupakan salah satu algoritma pengamanan text yang menggunakan kunci simetris yang dibuat oleh RSA Data *Security* Inc (RSADSI) yang berbentuk stream *cipher*. Pada aplikasi ini metode tersebut digunakan dalam data *QR Code* agar dapat menyembunyikan karakter khusus yang ditanam dalam *QR Code* tersebut. Hasil yang didapatkan dari penelitian ini bahwa metode RC4 berhasil diimplementasikan pada data *QR Code* dan kecepatan scanning tiap *QR Code* hanya berkisar 6 sampai 8 detik saja untuk setiap kelas. *QR Code* yang telah
ditanamkan metode tersebut hanya dapat didekripsi menggunakan sistem ini, sehingga meminimalisir kecurangan dalam proses absensi mahasiswa. (Muazharin Alfan, dkk, 2021).

1.1.3 Tinjauan Pustaka Literatur 3

Vigenere Cipher bekerja dengan melakukan enkripsi plainteks pada pesan dengan cara menggeser huruf pada pesan tersebut sejauh nilai kunci pada deret alphabet. Vigenère cipher adalah salah satu algoritma kriptografi klasik yang menggunakan metode substitusi abjadmajemuk. Substitusi abjad-majemuk mengenkripsi setiap huruf yang ada menggunakan kunci yang berbeda. Vigenere Cipher dibutuhkan untuk sistem ini agar hasil scan dari QR Code dapat dienkripsi dengan tujuan yaitu keamanan data. Hasil enkripsi nantinya akan memanggil suatu link agar nanti data kehadiran dapat ter-update pada database. (Nanda Dwi Wicaksono, 2019)

1.1.4 Tinjauan Pustaka Literatur 4

Penelitian ini menggunakan Android. Untuk keamanan datanya menggunakan Kriptografi Triple DES, dan menggunakan Caesar Cipher. Kriptografi Triple DES merupakan salah satu sitem pengamanan data yang sangat ketat. Dengan mengunakan caesar chiper data yang tersimpan di data base tetap terjaga Dengan melakukan enkripsi data dari hasil scan siswa menggunakan sistem ini lalu sistem memanggil link enkripsi agar dapat didekripsi dan dipilah dan dimasukkan kedalam database. Sedangkan QR Code merupakan sistem operasi mobile (OS) yang sangat populer dan banyak digunakan, Kode QR adalah sarana untuk memberikan informasi dengan cepat dan mendapatkan respon cepat tanpa input manual. Informasi yang dikodekan dalam kode QR dapat berupa alamat situs web, nomor telepon, pesan singkat, kartu nama, atau teks apa pun. (Prastyo Irwan Eka Susanto, dkk, 2021)

1.1.5 Tinjauan Pustaka Literatur 5

Penggunaan smart presensi yang dikombinasikan dengan teknologi *QR-CODE* dapat memberikan kepraktisan dan dapat memberikan solusi agar presensi berjalan dengan baik dan efisien. Pengawas ujian tidak lagi membubuhkan tanda tangan pada kartu ujian dan mahasiswa tidak lagi mencetak kartu ujian. Pemanfaatan fungsi dari smartphone akan memudahkan dosen dalam melakukan presensi secara online. Nomor Ujian dan NIM mahasiswa akan tersimpan dalam database dan akan ditampilkan menggunakan *QR-CODE*, saat pengawas ujian melakukan scanning *QR-CODE*, maka mahasiswa akan menyerahkan *QR-CODE* yang telah tercetak di Smartphone. Pengamanan data yang dilakukan adalah dengan memanfaatkan kode batang *QR-CODE* menggunakan enkripsi algoritma AES 256 Bit. Aplikasi Smart Presensi pada Ujian di Institut Teknologi Nasional Malang merupakan alternatif untuk mempermudah dan menyederhanakan proses presensi. (Zakaria Adi Putra, 2019)

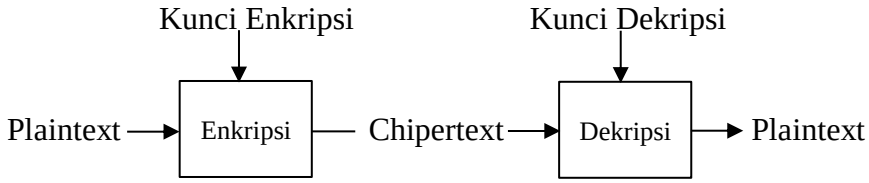
Dari Tinjauan pustaka diatas penulis menyimpulkan dan memilih Algoritma AES *(Advanced Encryption Standard)* untuk kasus Pengembangan Absensi dengan implementasi teknologi *QR-CODE,* dikarenakan algoritma AES *(Advanced Encryption Standard)* merupakan algoritma yang lebih aman, efisien, lebih cepat dan penggunaan memori lebih sedikit daripada

semua algoritma dengan memungkinkan ukuran kunci 256-bit dan melindungi dari serangan di masa depan (U. Thirupalu, 2019)

2.2 Sistem Enkripsi dan Dekripsi

Menurut Dan Boneh dan Victor Shoup (2015:18), "Kriptografi adalah kasus bagaimana dua pihak dapat berkomunikasi secara rahasia di internet dengan adanya kehadiran penyadap". Pada dasarnya kriptografi terdiri dari dua proses, yaitu proses Enkripsi dan Dekripsi. Proses Enkripsi adalah proses penyandian terbuka menjadi pesan rahasia*(Chipertext)*. Pada saat *Chipertext* diterima oleh penerima, maka pesan rahasia tersebut akan diubah melalui proses Dekripsi menjadi pesan terbuka sehingga pesan tersebut dapat dibaca oleh penerima pesan. (Lie Clara, 2021)

Enchiper merupakan proses perubahan teks asli *plaintext* data ke dalam bentuk kode. Kode dalam konteks ini merupakan sistem untuk menampilkan data dengan set karakter, angka, simbol, kata dan sinyal. Secara singkat proses enkripsi merupakan algoritma yang mengubah *plaintext* menjadi *chipertext* dengan menggunakan sebuah kunci. (Dony Arius, 2009).



Gambar 2.1 Diagram alir proses *Enkripsi* dan *Dekripsi*

2.2.1 Keunggulan dan Kelemahan Sistem Enkripsi

Jika dilihat dari definisinya tentu sistem atau proses enkripsi membawa banyak dampak positif. (Ijang, 2019). Berikut ini adalah keunggulan dan kelemahan dalam sebuah sistem enkripsi :

1. Sebuah data menjadi lebih aman dan hanya bisa diakses oleh pengguna yang mempunyai kunci.
2. Sebagai tanda bahwa suatu data merupakan data asli dan tidak dimanipulasi.
3. Mencegah terjadinya proses penyadapan data oleh pihak ketiga. (Ijang, 2019).
4. Mengatasi penyadapan pada media komunikasi.
5. Memberikan otentikasi dan perlindungan integritas. (Fitho, 2021)

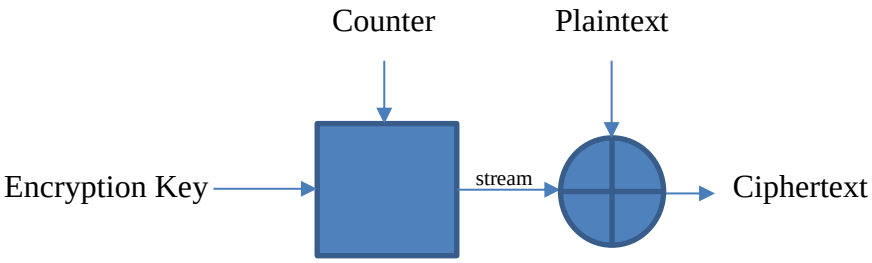
Namun adapun kelemahan dalam penggunaan enkripsi terhadap suatu data (Ijang, 2019). Antara lain :

1. Dapat digunakan untuk kegiatan kejahatan seperti komunikasi antar teroris
2. Dapat digunakan untuk menyimpan berbagai data kriminal.
3. Jika salah satu pengguna kehilangan kuncinya *(decryptor)*, maka data tersebut tidak akan bisa diDekripsi dan dibaca (Ijang, 2019).
4. Catatan kriminal disembunyikan oleh penjahat yang sudah memiliki *record* kejahatan yang banyak. (Fitho, 2021).

2.3 AES *(Advanced Encryption Standard)*

AES merupakan lanjutan dari algoritma enkripsi DES yang pada 2 maret tahun 2001 ditetapkanlah algoritma baru Rijndael sebagai AES. Kriteria pemilihan AES didasarkan pada 3 kriteria utama yaitu keamanan, harga, dan karakteristik algoritma beserta implementasinya.

AES termasuk dalam kriptografi yang sifatnya simetri dan *cipher block.* Oleh karena itu algoritma AES mempergunakan kunci yang sama saat enkripsi dan Dekripsi serta masukan dan keluarnya berupa blok dengan jumlah bit tertentu.



Gambar 2.2 Algoritma *AES*

AES mendukung berbagai variasi ukuran blok dan kunci yang akan digunakan. Namun AES mempunyai ukuran blok dan kunci yang tetap sebesar 128, 192, 256 bit.

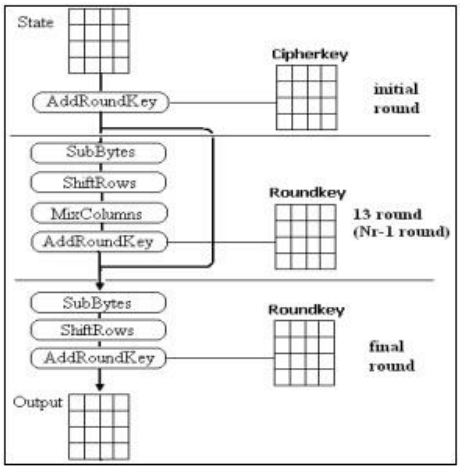
Tabel 2.2 Perbandingan Jumlah *Round* dan *Key*

	Jumlah Key (Nk)	Ukuran Block (Nb)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Pemilihan ukuran blok dan kunci akan menentukan jumlah proses enkripsi dan Dekripsi yang dilalui.

2.3.1 Proses Enkripsi Advanced Encryption Standard (AES)

Proses enkripsi pada algoritma AES terdiri dari 4 jenis transformasi *bytes*, yaitu *SubBytes*, *ShiftRows*, *MixColums*, dan *AddRoundKey*. Pada awal proses enkripsi, input yang telah disalinkan ke dalam state akan mengalamai transformasi byte AddRoundKey. Setelah itu State akan mengalami transformasi Subytes, ShiftRows, Mix Columns, dan AddRoundKey secara berulang sebanyak Nr. Proses ini disebut Round Function. Ilustrasi proses enkripsi AES dapat digambarkan seperti pada gambar dibawah ini



Gambar 2.3 Proses Enkripsi *AES*

A. Key Schedule

Proses key schedule diperlukan untuk mendapatkan subkey dari kunci utama agar cukup untuk melakukan enkripsi dan Dekripsi. Terdiri dari beberapa proses yaitu :

a. Operasi Rotate, merupakan operasi perputaran 8 bit pada 32 bit dari kunci.
b. Operasi SubBytes, pada operasi 8 bit dari subkey disubstitusikan dengan nilai S-Box.

c. Operasi Rcon, Operasi ini dapat diterjemahkan sebagai operasi pangkat 2 nilai tertentu dari *user*. Operasi ini menggunakan nilai-nilai dalam Galois field. Nilai-nilai dari Rcon kemudian akan di-XOR dengan hasil operasi *SubBytes*.
d. Operasi XOR dengan w[i-Nk] merupakan word yang berada pada Nk sebelumnya.

B. AddRoundKey

Pada proses enkripsi dan Dekripsi AES proses *AddRoundKey* sama, sebuah *round key* ditambahkan pada *state* dengan operasi XOR. Setiap *round key* terdiri dari Nb *word* dimana tiap *word* tersebut akan dijumlahakan dengan *word* atau kolom yang bersesuaian dari *state* sehingga :

$[S_{0,c}, S_{1,c}, S_{2,c}, S_{3,c},] = [S_{0,c}, S_{1,c}, S_{2,c}, S_{3,c},] \oplus [W_{round*Nb+c}]$ untuk $0 \leq c \leq Nb$

[w_i] adalah *word* dari *key* yang bersesuaian dimana i = round*Nb+c. Transformasi pada proses enkripsi pertama kali pada round = 0 untuk *round* selanjutnya *round* = *round* + 1, pada proses Dekripsi pertama kali pada *rounf* = 14 untuk *round* selanjutnya *round* = *round* – 1.

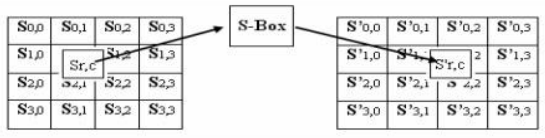
C. SubBytes

AES hanya memiliki satu S-Box. Kriteria desain untuk kotak S yang dibuat sedemikian rupa sehingga tahap terhadap diferensial linear yang dikenal sebagai pembacaan sandi dan menyerang menggunakan manipulasi aljabar. Koordinant X merupakan digit pertama sedangkan Y yang kedua dari bilangan *hexadecimal.*

		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

Gambar 2.4 S-Box SuBytes

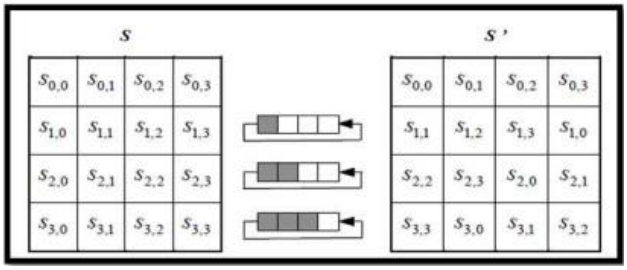
Untuk setiap *byte* pada array state, misalnya S[*r*,*c*] = xy, yang dalam hal ini *xy* adalah digit heksadesimal dari nilai S[*r*,*c*], maka nilai substitusinya, dinyatakan dengan S'[*r*,*c*], adalah elemen di dalam tabel substitusi yang merupakan perpotongan baris *x* dengan kolom *y*. Berikut gambar pengaruh pemtaan byte pada setia[nbyte dalam state



Gambar 2.5 Pengaruh Pemetaan pada setiap Byte dalam state

D. ShiftRows

Proses ShiftRows akan beroperasi pada tiap baris dari tabel state. Proses ini akan bekerja dengan cara memutar byte pada 3 baris terakhir (baris 1,2, dan 3) dengan jumlah perputaran yang berbeda-beda. Baris 1 akan diputar sebanyak 1 kali, baris 2 akan diputar sebanyak 2 kali, dan baris 3 akan diputar sebanyak 3 kali. Sedangkan baris 0 tidak akan diputar.



Gambar 2.6 Transformasi *ShiftRows*

E. MixColums

Proses MixColums akan beroperasi pada tiap kolom dari tabel state. Operasi ini menggabungkan 4 bytes dari setiap kolom tabel state dan menggunakan transformasi linier Operasi Mix Colums memperlakukan setiap kolom sebagai polinomial 4 suku dalam Galois field dan kemudian dikalikan dengan c(x) modulo(x4+1), dimana c(x)=3x3+x2+x+2. Kebalikkan dari polinominal ini adalah c(x)=11x3+13x2+9x+14. Operasi MixColums juga dapat dipandang sebagai perkalian matrix.

AES memiliki kemampuan untuk bekerja sangat baik pada platform apapun. Ditambah dengan operasi yang menggunakan table lookup dan operasi XOR membuat prosesnya menjadi tidak rumit. (Lie Clara, 2020).