BAB 2
LANDASAN TEORI

2.1 *Word Processor*

Word processor software atau perangkat lunak pengolah kata adalah perangkat lunak yang dibangun khusus untuk mengolah kata dan umumnya disajikan menggunakan teks [19]. Aplikasi berupa *word processor* sangat dibutuhkan dalam berbagai bidang kehidupan seperti pendidikan, sosial ekonomi, kesehatan, keuangan, dan lainnya [20]. Perangkat lunak atau *software word processor* dapat digunakan untuk membuat keperluan berbagai dokumen seperti surat, tesis, proposal, dan lainnya [21]. Salah satu contoh *software processor software* adalah Google Docs yang merupakan perangkat lunak pengolah kata yang dikembangkan oleh Google [22].

2.2 Enkripsi dan Dekripsi

Enkripsi merupakan proses mengubah atau menerjemah teks biasa (*plain text*) menjadi teks tersandi (*cipher text*). Proses enkripsi bisa dilakukan dengan menggunakan algoritma yang dirancang untuk melakukan enkripsi seperti AES (*Advanced Encryption Standard*), RSA (*Rivest–Shamir–Adleman*), dan lainnya. Dekripsi merupakan proses sebaliknya yaitu mengubah *cipher text* menjadi teks biasa atau *plain text*.

Terdapat dua jenis *cipher* yaitu *block cipher* dan *stream cipher* [23]. *Block cipher* merupakan *symmetric key cipher* yang beroperasi pada kelompok bit yang memiliki panjang tetap dan transformasinya tidak dapat berubah [24]. *Stream cipher* merupakan *symmetric key cipher* yang menggunakan aliran kunci yang sesuai untuk menghasilkan *cipher text* dalam proses enkripsi [25].

2.3 *Web Socket*

Web socket merupakan protokol yang membantu halaman *web* untuk komunikasi dua arah antar *client* dan *server* agar bisa berkomunikasi secara *real time* [26]. Koneksi *websocket* umumnya menggunakan *port* TCP 80 dan 443 agar dapat bekerja secara optimal. Setelah koneksi *web socket* dibuat maka perangkat yang terhubung dapat mengirim dan menerima data secara independen dan tanpa

permission apapun [27]. *Web socket* menggunakan HTTP sebagai mekanisme transport sehingga komunikasi antar *client* dan *server* tidak langsung berakhir ketika *client* menerima respon melainkan komunikasi masih tetap berlangsung secara *asynchronous* [28].

Terdapat *web socket library* yang berbasis JavaScript, salah satunya adalah Socket.IO. Socket.IO merupakan *library* yang memungkinkan komunikasi antara *client* dan *server* dengan latensi rendah, berlangsung secara dua arah, dan *event-based*. Koneksi Socket.IO dapat dibuat dengan tiga transport tingkat rendah berbeda yaitu HTTP *long-polling*, *WebSocket*, dan *WebTransport*. Socket.IO akan secara otomatis memilih transport terbaik yang tersedia berdasarkan kemampuan *browser* dan koneksi internet. Terdapat beberapa fitur pada Socket.IO yaitu HTTP *long polling fallback*, *automatic reconnection*, *packet buffering*, *broadcasting*, dan *multiplexing* [29].

2.4 *Advanced Encryption Standard* (AES)

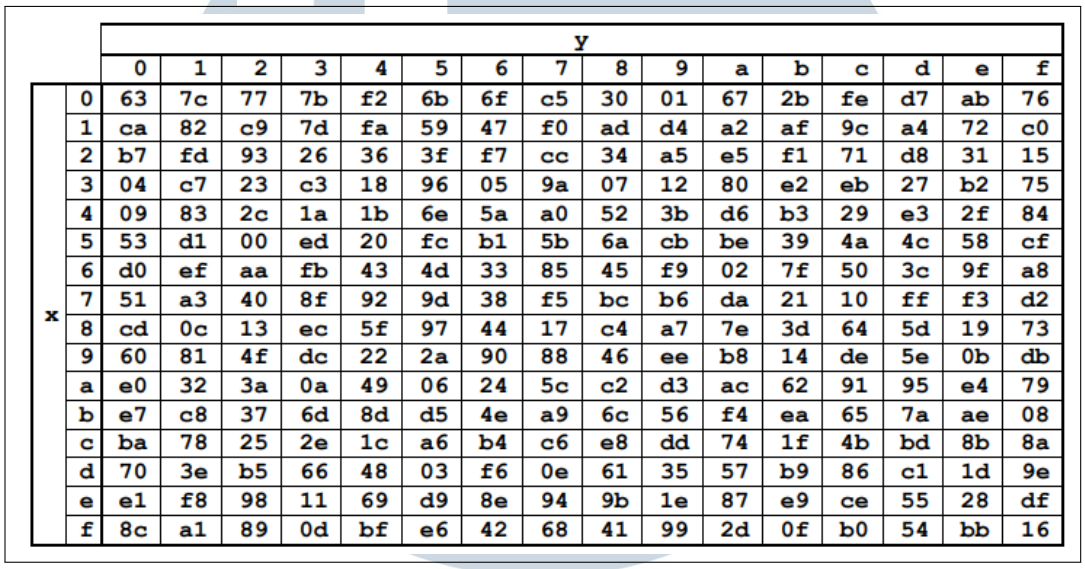
Advanced Encryption Standard (AES) merupakan algoritma simetris yang digunakan untuk melakukan enkripsi dan dekripsi sebuah data atau informasi [30]. Algoritma AES diperkenalkan pada tahun 2001 oleh National Institute of Standards and Technology [31]. AES termasuk ke dalam algoritma enkripsi *block cipher*. Dalam proses enkripsi dan dekripsi data dengan panjang kunci yang bervariasi yaitu 128 bit, 192 bit, dan 256 bit [31].

Terdapat dua proses utama dari proses melakukan enkripsi pada AES yaitu Key Expansion dan Rounds. Proses Key Expansion merupakan proses untuk merubah *key state* atau kunci awal sebanyak beberapa kali sesuai dengan ukuran kunci yang akan digunakan. Hasil yang didapatkan dari proses tersebut akan digunakan pada setiap perputaran saat proses Rounds. Pada ukuran kunci 128 bit akan memiliki 10 putaran, 192 bit akan memiliki 12 putaran, dan 256 bit akan memiliki 14 putaran [31].
Proses Key Expansion dapat dijabarkan sebagai berikut:

1. Proses RotWord, proses ini merupakan proses sistem mengambil kolom terakhir pada *key state* yang kemudian urutan baris dari akan dinaikkan satu. Baris pertama akan menjadi baris terakhir, baris kedua akan menjadi baris pertama, baris ketiga akan menjadi baris kedua, dan baris keempat menjadi baris ketiga. Contoh dari proses RotWord yaitu apabila nilai kolom terakhir

pada *key state* dalam bentuk heksadesimal adalah 09 cf 4f 3c akan diubah urutannya menjadi cf 4f 3c 09.

2. Proses SubWord merupakan proses untuk menukar nilai hasil yang didapatkan pada proses RotWord menjadi nilai yang sesuai pada S-Box. Bentuk dari tabel S-Box dapat dilihat pada gambar 2.1.



		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 2.1. S-Box *Advanced Encryption Standard*
sumber: NIST

Pada gambar dapat dilihat x melambangkan baris dan y melambangkan kolom atau karakter terakhir dari nilai heksadesimal yang didapatkan. Apabila nilai yang didapatkan dari hasil RotWord adalah cf 4f 3c 09 maka untuk menukar nilai 'cf' maka sistem akan mengambil nilai dari baris c kolom f yaitu dengan nilai '8a' dan seterusnya hingga '09'. Hasil akhir dari proses SubWord yang akan didapatkan yaitu 8a 84 eb 01.

3. Proses XOR nilai SubWord dengan tabel RCON, pada proses ini sistem akan melakukan proses XOR pada nilai hasil akhir SubWord dengan nilai yang ada pada tabel RCON. Nilai pada tabel yang akan digunakan pada RCON akan disesuaikan dengan jumlah putaran yang sedang berlangsung. Gambar 2.2 merupakan gambar dari tabel RCON.

   Pada gambar dilihat untuk putaran pertama memiliki nilai 01000000, nilai tersebut akan digunakan untuk melakukan proses XOR dengan hasil dari proses SubWord sehingga 8a 84 eb 01 XOR 01 00 00 00. Dari hasil XOR yang dilakukan didapatkan nilai 8b 84 eb 01.

i	Rcon[i]
1	01000000
2	02000000
3	04000000
4	08000000
5	10000000
6	20000000
7	40000000
8	80000000
9	1b000000
10	36000000

Gambar 2.2. Tabel RCON

4. Proses terakhir adalah melakukan XOR hasil proses sebelumnya dengan *key state* per kolom. Kolom pertama akan di-XOR dengan hasil proses XOR sebelumnya. Hasil dari XOR kolom pertama akan di-XOR dengan kolom kedua dan seterusnya hingga kolom terakhir. Proses ini merupakan akhir dari putaran sehingga sistem akan melakukan pengulangan dengan proses yang sama dengan mengambil kolom terakhir dari hasil XOR kolom terakhir dan kembali ke proses RotWord. Berdasarkan penjelasan sebelumnya, proses perulangan akan bergantung dengan ukuran dari *key* yang digunakan.

Proses utama selanjutnya adalah proses Rounds, proses ini adalah merupakan proses yang akan memperoleh hasil berupa nilai enkripsi terhadap *plain text*. Terdapat empat buah proses utama dari proses Rounds, pada proses ini sistem akan melakukan proses AddRoundKey terlebih dahulu, dan selanjutnya sistem akan melakukan pengulangan empat proses transformasi utama yaitu SubByte, ShiftRows, MixColumns, dan AddRoundKey sesuai dengan ukuran *key* yang digunakan. Pada proses putaran terakhir terdapat perbedaan yaitu sistem hanya akan melakukan tiga buah proses yaitu SubBytes, ShiftRows, dan AddRoundKey saja. Pada proses awal atau *initial round* sistem akan melakukan XOR terhadap *plain text* dengan *key state* yang digunakan. Berikut adalah penjelasan dari empat

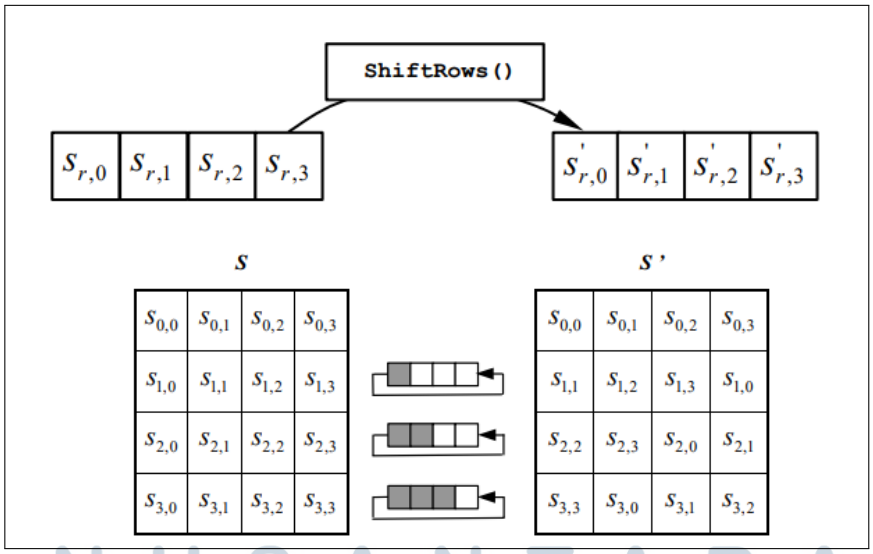
proses transformasi utama pada proses Rounds.

1. *SubBytes*

   Transformasi *SubBytes* adalah substitusi byte non-linear yang beroperasi secara mandiri atau independen pada setiap byte pada state menggunakan *S-box*. S-box yang digunakan pada proses SubBytes menggunakan S-box yang sama dengan proses SubWord yang dapat dilihat pada gambar 2.1. Hasil dari XOR pada *initial round* dapat dimisalkan S_i, i memiliki nilai 75 dan nilai substitusi akan ditentukan dari nilai tersebut. Digit pertama pada nilai S_i, i akan menentukan baris dari S-box dan digit kedua akan menjadi penentu kolom untuk nilai substitusi menggunakan S-box. Dapat dilihat bahwa baris 7 dan kolom 5 memiliki nilai "9d" yang merupakan nilai dari hasil transformasi atau subtitusi. Proses ini dilakukan pada semua nilai yang ada pada hasil XOR sebelumnya yang didapatkan dari hasil *initial round* [30].

2. *ShiftRows*

   Pada proses ini yaitu proses *ShiftRows* merupakan proses transformasi yang melakukan penggeseran menggunakan *offset* yang berbeda-beda pada 3 baris *state* terakhir. Pada transformasi *ShiftRows* baris pertama tidak akan digeser menggunakan *offset*.



Gambar 2.3. Proses transformasi *ShiftRows*
sumber: NIST

Dalam proses transformasi pergeseran dilakukan sesuai nilai r pada masing-masing baris $(S_r, 4)$. Pada baris kedua yaitu S_1 dilakukan pergeseran atau

shifting sebanyak satu *byte* ke kiri. Pada baris ketiga akan digeser ke kiri sebanyak dua *byte* pada matriks. Terakhir melakukan pergeseran tiga *byte* ke kiri pada baris terakhir matriks [30].

3. *MixColumns*

Pada proses *MixColumns* merupakan proses operasi pencampuran kolom per kolom yang setiap elemen pada kolom akan dilakukan perkalian matriks dan dilakukan operasi logika XOR.

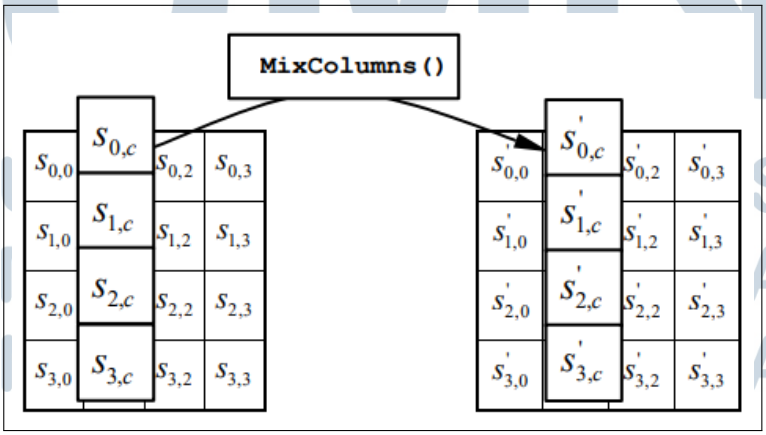
Berikut adalah rumus perkalian matriks yang dapat ditulis,

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \leq \boldsymbol{Nb} \tag{2.1}$$

Dari hasil perkalian matriks, empat kolom atau *bytes* pada matriks dapat diganti atau direpresentasikan sebagai berikut:

$$\begin{aligned} s'_{0,c} &= (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c} \\ s'_{1,c} &= s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c} \\ s'_{2,c} &= s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c}) \\ s'_{3,c} &= (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}) . \end{aligned} \tag{2.2}$$

Proses operasi *MixColumns* dapat diilustrasikan seperti pada gambar 2.4

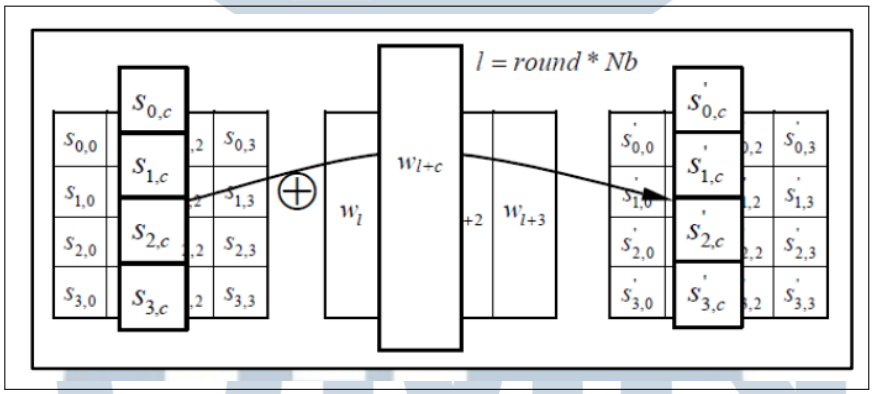


Gambar 2.4. Ilustrasi operasi *MixColumns*
sumber: NIST

Pada gambar 2.4 dapat dilihat bahwa pada proses MixColumns untuk proses perkalian dilakukan dengan mengalikan setiap baris pada matriks baku atau matriks yang sudah ditentukan pada proses enkripsi dengan masing-masing kolom pada hasil yang didapatkan melalui proses ShiftRows. Rumus dan gambar yang ada pada penjelasan sebelumnya merupakan ilustrasi dari perkalian untuk mendapatkan nilai kolom kedua pada *state* [30].

4. *AddRoundKey*

Pada transformasi *AddRoundKey*, sebuah *round key* ditambahkan pada setiap *state* menggunakan operasi *bitwise* XOR sederhana. Proses operasi XOR pada proses AddRoundKey ini merupakan operasi antara hasil yang didapatkan pada proses MixColumns dengan nilai yang didapatkan pada proses Key Expansion sesuai dengan jumlah putaran yang ada. Pada putaran pertama operasi XOR akan antara proses MixColumns pertama dengan hasil Key Expansion pertama dan selanjutnya. Gambar 2.5 merupakan ilustrasi dari transformasi *AddRoundKey*,



Gambar 2.5. Ilustrasi transformasi *AddRoundKey*
sumber: NIST

Proses AddRoundKey ini merupakan proses akhir dari setiap putaran yang dilakukan dalam proses Rounds. Perputaran akan berlangsung sesuai dengan ukuran *key* yang digunakan. Berdasarkan penjelasan sebelumnya, pada ukuran *key* 128 bit akan dilakukan 10 putaran, 192 bit akan dilakukan 12 putaran, dan terakhir pada 256 bit akan dilakukan sejumlah 14 putaran. Pada setiap putaran terakhir proses MixColumns akan dilewati sesuai dengan aturan yang ada pada AES [30].

2.5 *Two Factor Authentication*

Two factor authentication atau 2FA merupakan sebuah metode otentikasi yang menggunakan 2 faktor independen untuk membuktikan bahwa sebuah entitas asli atau tidak [7]. 2FA dapat mencegah atau mengurangi resiko terhadap seorang penyerang yang dapat masuk ke dalam sistem karena mengetahui kata sandi milik *user*. Hal ini disebabkan karena dengan adanya 2FA seorang penyerang harus mengetahui atau dapat melakukan verifikasi identitasnya [7]. Hal ini dapat dikatakan bahwa pengguna tidak hanya akan memasukkan kata sandi saja melainkan harus melewati satu tahap verifikasi identitas lain seperti OTP (*One Time Password*), *biometric*, dan lainnya [32].

One time password merupakan metode otentikasi yang pada prosesnya menggunakan kode yang bersifat dinamis atau dapat berubah pada interval waktu tertentu [33]. Kode OTP memiliki umumnya terdapat 4 sampai 6 digit yang terdiri dari huruf dan angka. Metode ini melakukan perubahan kode pada rentang waktu tertentu untuk faktor keamanan yaitu untuk menghindari serangan seperti *brute force* [34]. Sifat OTP yang dinamis membuat keamanan pada sistem dikarenakan apabila penyerang telah berhasil merekam kode OTP sebelumnya maka kode yang terekam tidak dapat digunakan dua kali atau lebih [35]. Umumnya, kode OTP dapat dikirimkan sistem kepada pengguna melalui via SMS (*short messaging service*), Whatsapp, ataupun melalui *email* [36].

2.6 Website

Website atau situs web adalah kumpulan dari beberapa laman yang berisi informasi baik dalam bentuk teks, gambar, video, ataupun audio yang terhubung ke dalam internet [37]. Terdapat 2 jenis dalam *website* berdasarkan sifatnya, yaitu [38]:

1. Website Statis

   Website statis merupakan *website* yang memiliki halaman statis atau tidak dapat dirubah. Halaman pada *website* statis dapat dirubah tetapi harus dilakukan perubahan dari kode secara manual. Hal ini mengartikan bahwa informasinya biasa hanya satu arah atau hanya dapat dirubah oleh pemiliknya [38].

2. Website Dinamis

Website dinamis merupakan situs web yang memiliki halaman yang selalu *update* atau dapat berubah dan pada situs web yang bersifat dinamis biasanya memiliki halaman *backend* atau administrator. Situs web dinamis memiliki informasi dua arah yaitu berasal dari pengguna dan pemilik *website* [38].

Halaman pada *website* biasanya dibangun menggunakan bahasa standar HTML (*Hypertext Markup Language*) yang diterjemahkan oleh *browser* sehingga mudah terbaca oleh pengguna [39]. *Hypertext Markup Language* atau HTML adalah bahasa standar yang digunakan untuk menampilkan konten seperti gambar, video, audio, dan lainnya pada halaman situs web [40]. Dalam pengembangan sebuah *website* terdapat dua bagian metode yaitu *backend* dan *frontend* [41].

Backend merujuk pada lingkungan komputasional dimana operasi dan proses yang berkaitan dengan suatu sistem informasi atau aplikasi dijalankan. Pada bagian ini, dilakukan penambahan, pengubahan, dan penghapusan data sesuai dengan kebutuhan sistem [42]. *Frontend* adalah pengembangan antarmuka pada *website* menggunakan beberapa bahasa standar seperti HTML, CSS, dan *Javascript* sehingga pengguna dapat melihat dan berinteraksi pada halaman *website* tersebut [43]. Pengembangan bagian *website* seperti *backend* dan *frontend* umumnya dibantu dengan bantuan kerangka kerja (*framework*).

Salah satu *framework* pada bagian *backend* adalah *framework* Laravel. Laravel merupakan *framework* pada bagian *backend* yang paling popular pada saat ini. Laravel adalah salah satu *framework* bahasa pemrograman PHP yang bersifat *open source* dengan menggunakan pola Model-View-Controller (MVC) [44]. Laravel dibangun oleh Taylor Otwell pada tahun 2011 dan untuk versi Laravel sudah mencapai versi 10 pada tahun 2024 [45]. Berlainan pada *frontend*, *framework frontend* paling popular adalah React.

React merupakan *library* dari JavaScript untuk membuat *user interface* atau antarmuka pengguna yang bersifat *open source* [46]. Salah satu kelebihan dari *framework* React adalah *single-page application* (SPA) [47]. *Single-page application* merupakan aplikasi web yang terdiri dari banyak komponen individual yang dapat diganti secara terpisah tanpa harus melakukan *refresh* pada halaman [48]. Dalam *website* tentunya tidak terlepas dari sistem *database* atau basis data. Sistem basis data merupakan sekumpulan data-data yang disajikan dalam bentuk tabel dan dalam bentuk *field* dan kolom-kolom [49]. Salah satu perangkat lunak dari *database* adalah MySQL. MySQL merupakan *open source* SQL *database* yang paling popular yang dikembangkan dan didistribusikan oleh Oracle Companion [50].

Dalam mengembangkan *online word processor* diperlukan *library* yang dapat membantu dalam pembuatan salah satunya adalah QuillJS. QuillJS merupakan *library rich text editor* yang bersifat *open source what you see is what you get* editor yang dapat digunakan secara gratis [51].

2.7 *Brute Force Attack*

Brute Force attack adalah serangan siber yang dilakukan dengan mencoba setiap kombinasi sebuah kunci atau kata sandi untuk mendapatkan akses yang tidak sah [52]. Teknik penyerangan ini bergantung pada kecepatan dan ketelitian komputer untuk melakukan percobaan setiap kombinasi dari kata kunci yang ada. Cara kerja pada teknik *brute force* sendiri adalah mencoba segala kombinasi yang ada. Salah satu contohnya terhadap empat buah karakter kode maka peretas akan mulai melakukan penyerangan dari 0000 hingga 9999 dan peretas akan mendapatkan akses dalam 10000 percobaan [53]. Semakin banyak jumlah karakter yang ada pada kode maka probabilitas atau jumlah percobaan yang dibutuhkan akan semakin kompleks dan memakan waktu yang signifikan lebih lama.

2.8 Avalanche Effect

Avalanche Effect merupakan suatu metode untuk mengetahui perubahan pada saat melakukan enkripsi dalam persen dengan membandingkan rasio jumlah perbedaan bit dengan jumlah total bit pada *cipher text*. Semakin tinggi presentase yang didapatkan menandakan semakin baik hasil yang didapatkan melalui proses enkripsi. Hasil presentase yang didapatkan harus memiliki perubahan atau efek di atas 50% untuk dapat dikatakan sebagai enkripsi yang baik. Efek yang didapatkan memastikan penyerang tidak dapat menebak atau memprediksi teks dengan mudah melalui analisis statistik [54]. Berikut adalah rumus dari perhitungan Avalanche Effect.

$$\text{Avalanche Effect} = \frac{\text{jumlah bit berbeda}}{\text{total bit}} \times 100\% \tag{2.3}$$