BAB 2
TINJAUAN PUSTAKA

2.1 Profil Puslitbang Geologi Kelautan

Profil instansi Puslitbang Geologi Kelautan meliputi sejarah berdirinya, visi dan misi serta struktur organisasi.

2.1.1 Sejarah Puslitbang Geologi Kelautan

Pusat Penelitian dan Pengembangan Geologi Kelautan (P3GL) adalah salah satu unit yang bernaung di bawah Badan Penelitian dan Pengembangan Kementerian Energi dan Sumber Daya Mineral (Balitbang ESDM) mempunyai tugas-tugas melaksanakan penelitian, pengembangan, perekayasaan, pengkajian, survei dan pemetaan bidang geologi kelautan. antara lain:

1. Penyiapan penyusunan kebijakan teknis, rencana dan program penelitian, pengembangan dan perekayasaan, pengkajian, survei dan pemetaan di bidang geologi kelautan;
2. Pelaksanaan penelitian, pengembangan, perekayasaan, pengkajian, survei dan pemetaan, serta pengelolaan pengetahuan dan inovasi di bidang geologi kelautan
3. Pemantauan, evaluasi dan pelaporan pelaksanaan penelitian, pengembangan, dan perekayasaan, pengkajian, survei dan pemetaan di bidang geologi kelautan
4. Pelaksanaan administrasi Pusat Penelitian dan Pengembangan Geologi Kelautan.

Adapun unit kerja yang berada di bawah Badan Penelitian dan Pengembangan Kementerian Energi dan Sumber Daya Mineral (Balitbang ESDM) selain P3GL adalah Puslitbang Teknologi Mineral dan Batubara (TEKMIRA), Puslitbang Teknologi Minyak dan Gas Bumi (LEMIGAS), Puslitbang Ketenagalistrikan, Energi Baru, Terbarukan dan Konservasi Energi (P3TKEBTKE), dan dukungan manajemen Balitbang ditangani oleh Sekretariat Badan Litbang ESDM.

2.1.2 Visi dan Misi Puslitbang Geologi Kelautan (P3GL)

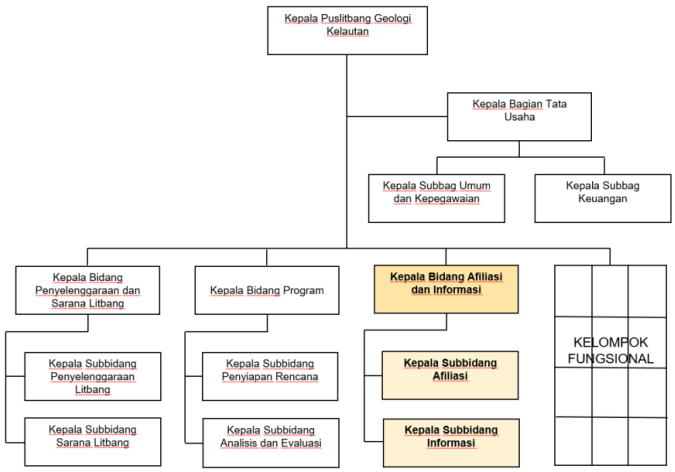
P3GL memiliki visi yaitu: Menjadi pusat penelitian dan pengembangan geologi kelautan yang profesional, unggul, dan mandiri di bidang energi dan sumber daya mineral. Untuk mewujudkan visi tersebut, P3GL menjalankan misi antara lain:

1. Melaksanakan pemetaan geologi kelautan bersistem di seluruh wilayah perairan Laut Indonesia.
2. Melaksanakan litbang potensi energi dan sumber daya mineral kawasan pesisir dan laut.
3. Melaksanakan litbang lingkungan dan kebencanaan geologi kelautan.
4. Melaksanakan kajian dalam perumusan, evaluasi dan rekomendasi kebijakan terkait dengan isu strategis nasional terkini.
5. Meningkatkan kualitas dan akses informasi sumber daya energi dan mineral kelautan, lingkungan dan kebencanaan geologi kelautan.
6. Melaksanakan pelayanan jasa teknologi di bidang geologi kelautan.
7. Mengembangkan program kegiatan, sumber daya manusia, sarana dan prasarana litbang di bidang geologi kelautan.Geologi.

2.1.3 Struktur Organisasi Puslitbang Geologi Kelautan (P3GL)

Susunan organisasi P3GL terdiri atas (Gambar 2.1):

1. Bagian Tata Usaha
2. Bidang Program
3. Bidang Penyelenggaraan dan Sarana Penelitian dan Pengembangan
4. Bidang Afiliasi dan Informasi
5. Kelompok Jabatan Fungsional



Gambar 2.1 Struktur Organisasi Puslitbang Geologi Kelautan

2.2 Landasan Teori

Landasan Teori bertujuan memberikan gambaran dari teori yang terkait dalam pembangunan aplikasi. Landasan Teori yang dibahas yaitu Pengertian Berkas Digital, Kriptografi, Metode yang digunakan dan Bahasa Pemrograman yang digunakan.

2.2.1 Pengertian Berkas Digital

Berkas digital (*file*) adalah identitas dari data yang disimpan di dalam sistem berkas yang dapat diakses dan diatur oleh pengguna. Sebuah berkas memiliki nama yang unik dalam direktori di mana ia berada. Alamat direktori dimana suatu berkas ditempatkan diistilahkan dengan *path* [3].

2.2.1.1 Jenis Berkas Digital

Berkas digital (*file*) dapat digolongkan menurut jenisnya, diantaranya adalah:

1. *File* Induk

   Suatu *file* yang diperlukan untuk memperlancar operasi sistem dan diperbaharui secara teratur.

2. *File* Transaksi

   Digunakan untuk memperbaharui *file* induk dengan informasi yang baru.

3. *File* Penunjang

   Merupakan kutipan dari file untuk menjaga kemungkinan adanya file asli yang rusak.

4. *File* Riwayat Hidup

   Informasi yang dikumpulkan selama periode waktu tertentu dan biasanya digunakan untuk menyusun laporan statistik.

5. *File* Data Transaksi

   Merupakan sebuah rekaman pada pita atau piringan untuk setiap transaksi yang melengkapi *file* induk.

6. *File* Kesalahan

   Rekaman tentang kesalahan yang disimpan pada file untuk pembentulan di kemudian hari.

7. *File* Laporan

   Merupakan turunan laporan tercetak yang ditahan pada piringan atau pita menunggu *printer* siap mencetak.

8. *File* Sementara

   File yang disiapkan untuk memproses peralihan.

9. *File* Pustaka (*Library*)

   File yang digunakan untuk menyimpan program aplikasi, program utiliti, dan program lain.

10. *File* Kerja (*Work*)

    File ini berisi *record-record* yang disusun sedemikian rupa sehingga dapat dibuat sebuah program dan dipakai oleh program lain sebagai *input*.

11. *File* Program

    File ini berisi perintah-perintah untuk memproses data. Perintah dapat ditulis dalam bahasa pemrograman, bahasa rakitan atau bahasa mesin.

2.2.1.2 Ekstensi Berkas Digital

Setiap berkas digital dalam media penyimpanan memiliki tanda pengenal atau ciri-ciri yang menyatakan jenis berkas tersebut. Umumnya pengenal tipe berkas tertera pada nama berkas tersebut, yaitu tiga huruf paling kanan setelah titik. Fungsinya adalah untuk mengetahui atau membedakan jenis berkas. Berikut ini adalah contoh macam-macam ekstensi berkas digital ditunjukkan pada Tabel 2-1:

Tabel 2.1 Ekstensi Berkas Digital

Ekstensi	Jenis	Aplikasi yang digunakan
001, 002, ...	Bagian-bagian dari suatu *file* tunggal	FFSJ, HJSplit, JJSplit, split (Unix)
MP4	*File video* untuk perangkat *mobile phone*	*Media Player Classic*
DOC, DOCX	Dokumen MS *Word*	*Microsoft Word* dan semua aplikasi pengolah kata
HTM, HTML	Dokumen *internet*	*Internet Explorer, Mozilla Firefox,* dan semua aplikasi penjelajah
JAVA	*Source code* bahasa pemrograman *Java*	NetBeans IDE dan semua aplikasi editor teks
JPEG, JPG	*File* gambar	*Paint, Microsoft Office Picture Manager*, dan semua aplikasi pengolah gambar

2.2.2 Pengertian Kriptografi

Kriptografi (*cryptography*) berasal dari bahasa Yunani yaitu *cryptos* artinya rahasia (*secret*) dan *graphein* artinya tulisan (*writing*). Jadi kriptografi berarti tulisan rahasia (*secret writing*). Kriptografi, secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita [9]. Selain pengertian tersebut, kriptografi merupakan ilmu yang mempelajari teknik-teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan data, keabsahan data, integritas data, serta autentikasi data [4]. Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi, yaitu:

1. Kerahasiaan, adalah 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, adalah 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, adalah 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-repudiasi atau nirpenyangkalan, adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman/terciptanya suatu informasi oleh yang mengirimkan/membuat.

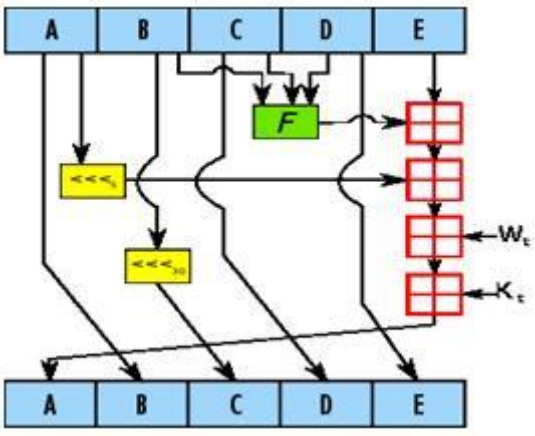
2.2.3 Fungsi *Hash*

Fungsi *Hash* digunakan untuk menjamin data atau file yang dikirim dan tidak mengalami modifikasi, pemalsuan atau injeksi selama transmisi (Message Integrity). Suatu fungsi hash akan memetakan bit-bit string dengan panjang sembarang ke sebuah string dengan panjang tertentu misal n. Proses pemetaan suatu input string

output tersebut disebut dengan proses hashing. Output dari fungsi hash disebut dengan nilai hash, kode hash atau hasil hash.

Cara kerja kriptografi algoritma SHA-1 adalah menerima input berupa pesan dengan ukuran sembarang dan menghasilkan message digest yang memiliki panjang 160 bit. Langkah-langkah pembuatan message digest dengan algoritma SHA-1 adalah sebagai berikut [5]:

1. Input Pesan yang akan di hash SHA-1.
2. Ubah pesan menjadi deretan biner
3. Penambahan Bit-bit pengganjal, yaitu dengan menambahkan pesan dengan sejumlah bit pengganjal sedemikian sehingga panjang pesan (dalam satuan bit) kongruen dengan 448 mod 512.
4. Penambahan nilai panjang pesan semula, yaitu pesan ditambah lagi dengan 64 bit yang representasi biner dari panjang pesan asli.
5. Inisialisasi Nilai Hash, pada algoritma SHA-1 nilai hash, H(0) terdiri dari 5 words dengan besar 32 bit dalam notasi hexadecimal.
6. Output nilai hash adalah nilai terakhir dari buffer. Berdasarkan tahapan yang ada pada Fungsi Hash SHA-1, maka skema Fungsi Hash SHA-1 dapat dilihat pada gambar berikut ini:



Gambar 2.2.3 Skema Fungsi Hash SHA-1

2.2.4 Algoritma *Advanced Encryption Standard (AES)*

AES (*Advanced Encryption Standard*) menggunakan algoritma Rijndael yang telah memenangkan sayembara terbuka yang dilakukan oleh NIST (*National Institute of Standard and Technology*). Sayembara ini dilakukan untuk menemukan algoritma baru untuk menggantikan algoritma DES (*Data Encryption Standard*) yang dirasa sudah tidak aman lagi [6]. NIST memberikan spesifikasi AES yaitu harus memiliki panjang blok 128 *bit* dan mampu mendukung panjang kunci 128, 192, dan 256. Setelah melalui beberapa tahapan seleksi, akhirnya NIST memilih sistem penyandian Rijndael yang dikembangkan oleh Joan Daemen dan Vincent Rijment sebagai sistem penyandian AES pada tahun 2000. Pemilihan Rijndael sebagai pemenang sayembara tersebut berdasarkan pada kriteria berikut ini:

1. Keamanan.

   Sistem penyandian harus tahan terhadap serangan analisis sandi selain serangan secara *brute force.*
2. Biaya.

   Sistem penyandian harus memiliki biaya komputasi dan memori yang efisien sehingga dapat diimplementasikan secara perangkat keras maupun perangkat lunak.
3. Karakteristik algoritma dan implementasi.

   Sistem penyandian harus bersifat terbuka, fleksibel, dan sederhana. Pada tahun 2001 akhirnya NIS memublikasikan AES sebagai standar pemrosesan dokumen pada dokumen FIPS-PUB 197 [8](NIST, 2001). AES merupakan algoritma *chiper* blok yang menggunakan teknik substitusi, permutasi dan sejumlah putaran pada setiap blok yang akan dienkripsi. Sistem permutasi dan substitusi (*S-box)* yang digunakan pada AES tidak menggunakan jaringan Feistel sebagaimana *chiper* blok pada umumnya [7].

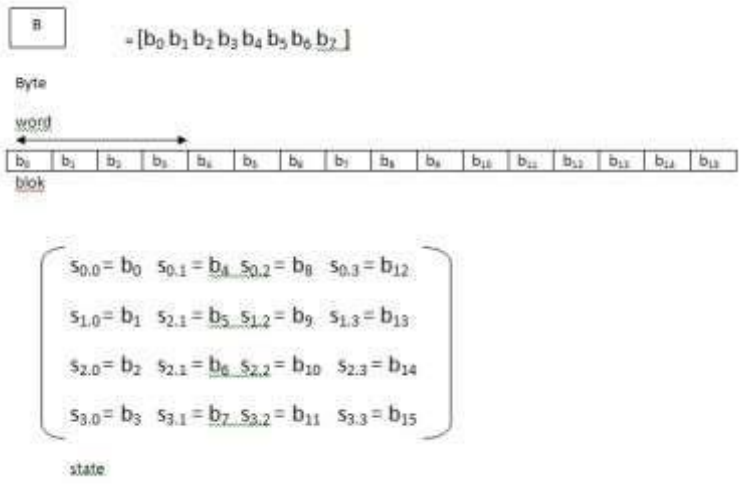
Tabel 2.1 Perbandingan jumlah kunci dan ronde AES

Panjang Kunci AES (*bit*)	Jumlah Ronde (Nr)
128	10
192	12
256	14

Algoritma Rijndael mempunyai tiga parameter yaitu:

1. Plainteks, *array* yang berukuran 16 *byte*, yang berisi data masukan.
2. Cipherteks, *array* yang berukuran 16 *byte*, yang berisi hasil enkripsi.
3. Kunci, *array* yang berukuran 16 *byte*, yang berisi kunci *cipher*.

AES memiliki panjang blok 128 *bit*. Kunci AES dapat memiliki panjang kunci *bit* 128, 192, dan 256 *bit*. Selain itu AES menggunakan 5 unit ukuran data : *bit*, *byte*, *word*, blok, *dan state*. *Bit* merupakan satuan data terkecil, yaitu nilai digit sistem biner. Sedangkan *byte* berukuran 8 *bit*, *word* berukuran 4 *byte* (32 *bit*), blok berukuran 16 *byte* (128 *bit*). Sedangkan *state* adalah blok yang ditata sebagai matriks *byte* berukuran 4x4 [9](Sadikin, 2012). Unit data AES dapat dilihat pada Gambar 2.2.



Gambar 2.2 Unit Data AES [9]

Blok *cipher* memiliki sifat bahwa setiap blok harus memiliki panjang yang sama (misalnya 128 *bit*). Apabila pada suatu blok AES memiliki panjang blok yang tidak sama, maka diperlukan mekanisme *padding. Padding* adalah penambahan *bitbit dummies* untuk menggenapi menjadi panjang blok yang sesuai. *Padding* biasanya dilakukan pada blok terakhir *plaintext*. *Padding* bada blok terakhir bisa dilakukan dengan berbagai macam cara, misalnya dengan penambahan *bit-bit* tertentu. Salah satu contoh penerapan *padding* dengan cara menambahkan jumlah total *padding* sebagai *byte* terakhir pada blok terakhir *plaintext*. Misalnya panjang blok adalah 128 *bit* (16 *byte*) dan pada blok terakhir terdiri dari 88 *bit* (11 *byte*) sehingga jumlah *padding* yang diperlukan adalah 5 *byte*, yaitu dengan menambahkan angka nol sebanyak 4 *byte*, kemudian menambahkan angka 5 sebanyak satu *byte*. Cara lain dapat juga menggunakan penambahan karakter *end-of-fil*e pada *byte* terakhir lalu diberi *padding* setelahnya.

Proses awal enkripsi AES dimulai dengan mengorganisir teks asli yang sudah diubah menjadi bilangan heksadesimal di dalam blok (128 *bit*) terlebih dahulu sebagai *state* yang berupa matriks (*array)* berukuran 4x4. *Chiper key* juga diorganisir menjadi matriks berukuran 4x4. Garis besar Algoritma AES yang beroperasi pada blok 128 *bit* dengan kunci 128 *bit* (di luar proses pembangkitan kunci AES) adalah sebagai berikut:

a. AddRoundKey

Melakukan XOR antara *state* awal dengan *chiper key* seperti yang terdapat pada Gambar 2.3. Tahap ini disebut juga dengan *initial round.*

49	49	49	49		2b	28	ab	09
44	44	44	44	XOR	7e	ae	f7	cf
49	49	45	59		15	d2	15	4f
30	33	34	36		16	a6	88	3c

Gambar 2.3 Transformasi *AddRoundKey*

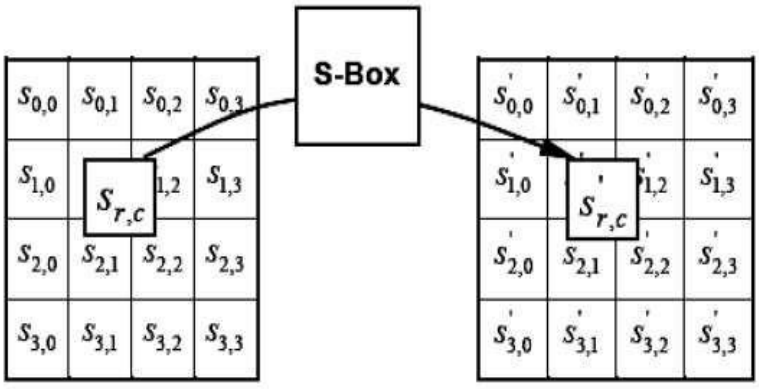
b. SubBytes

AES menggunakan substitusi nonlinear pada ukuran *byte* yang disebut dengan *SubBytes. SubBytes* mensubstitusikan satu *state* pada tabel tabel substitusi (S-Box) yang telah ditentukan nilainya. Tabel substitusi untuk *SubBytes* diberikan oleh Tabel 2.2.

Tabel 2.2 Tabel substitusi untuk transformasi *SubBytes* (S-Box)

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**e**	**f**
0	63	7c	77	7b	f2	6b	6f	c5	30	1	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	4	c7	23	c3	18	96	5	9a	7	12	80	e2	eb	27	b2	75
4	9	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	0	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	2	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	6	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	8
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	3	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

Misal untuk menggantikan nilai 88 pada *state*, maka yang bersesuaian dengan S-Box terletak pada baris 8 dan kolom ke 8, dan seterusnya dengan proses yang sama sampai 16 blok seperti Gambar 2.4.



Gambar 2.4 Ilustrasi *SubBytes*

c. ShiftRows

AES menggunakan permutasi untuk mengganti nilai pada elemen *state.* Permutasi ini hanya mengubah posisi elemen pada state tanpa mengubah nilainya. Transformasi permutasi pada *state* disebut dengan transformasi *ShiftRows. ShiftRows* dilakukan dengan cara memutar elemen matriks hasil proses transformasi *SubByte* pada baris 1, 2, dan 3 ke kiri dengan jumlah perputaran yang berbeda-beda. Baris pertama akan diputar sebanyak 1 kali, baris kedua sebanyak 2 kali, dam baris ke 3 akan diputar sebanyak 3 kali. Sedangkan baris ke 0 tidak diputar. Transformasi *ShiftRows* terlihat sederhana jika dilihat melalui representasi *state.* Namun menjadi rumit jika dilihat dari sudut pandang blok dikarenakan *state* merupakan representasi blok dengan orientasi per kolom.

d. MixColumns

MixColumns merupakan transformasi dengan cara mengalikan empat angka kolom *state* dalam GF *(Galois Field)* milik Rijndael (2^8).

Gambar 2.5 Transformasi *MixColumns*

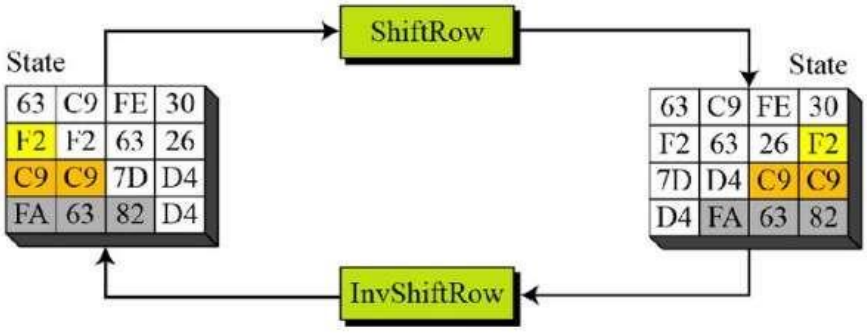
e. AddRoundKey

Setelah proses transformasi *MixColumns,* maka terdapat proses *AddRoundKey* dengan cara yang sama seperti sebelumnya, namun proses XOR nya dengan sub-kunci yang bersesuaian tiap iterasi. Algoritma dekripsi AES secara ringkas merupakan kebalikan dari algoritma enkripsi AES. Algoritma dekripsi AES menggunakan transformasi *invers* semua transformasi dasar yang digunakan pada algoritma enkripsi AES. Setiap transformasi dasar AES memiliki transformasi *invers*, yaitu:

1. *InvShiftRows*

InvShiftRows adalah transformasi *byte* yang berkebalikan dengan transformasi *ShiftRows.* Pada proses transformasi *InvShiftRows* dilakukan penggeseran *bit* ke kanan, sedangkan pada *ShiftRows* dilakukan penggeseran *bit* ke kiri. Pada baris kedua penggeseran *bit* dilakukan tiga kali, sedangkan pada baris ketiga dan baris keempat, dilakukan penggeseran *bit* sua kali dan satu kali [7](Setyaningsih, 2015).

Proses *InvShiftRows* dapat dilihat pada Gambar 2.3.



Gambar 2.6 Transformasi *InvShiftRows* [7](Setyaningsih, 2015)

2. *InvSubBytes*

InvSubBytes juga merupakan transformasi *byte* yang berkebalikan dengan transformasi *SubBytes.* Proses *InvSubBytes* dilakukan dengan cara setiap elemen pada *state* dipetakan dengan menggunakan tabel *inverse S-box.* Tabel ini berbeda dengan tabel *S-box* karena hasil yang didapat dari tabel ini adalah hasil dari dua proses yang berbeda urutannya, yaitu transformasi afin terlebih dahulu, kemudian perkalian *inverse* dalam GF(2^8). *Inverse S-box* dapat dilihat pada Tabel 2.3.

Tabel 2.3 *Inverse S-box* [6](Munir, 2006)

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

3. *InvMixColumns*

Setiap kolom dalam *state* dikalikan dengan matriks perkalian dalam AES. Perkalian dalam matriks dapat dilihat pada Gambar 2.4.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

Gambar 2.7 *InvMixColumns* [7](Setyaningsih, 2015)

4. *Inverse AddRoundKey*

Transformasi ini merupakan transformasi yang bersifat *self-invers* dengan syarat menggunakan kunci yang sama [9](Sadikin, 2012).

Sandi AES sampai saat ini masih dianggap aman untuk digunakan. Keamanan sistem sandi AES salah satunya disebabkan oleh penggunaan kunci yang besar (128 *bit*, 192 *bit*, dan 256 *bit*) apabila dibandingkan dengan sistem sandi DES yang hanya menggunakan 64 *bit*. Jadi *bruce attack* terhadap sistem sandi AES 256 *bit* memiliki ruang kunci 2^{256} yang merupakan nilai yang sangat besar [9](Sadikin, 2012).

2.2.3 Unified Modeling Language (UML)

Unified Modeling Language (UML) adalah bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. *Unified Modeling Language* (UML) adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi obyek (OOP) serta aplikasinya. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat tool untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh Object Management Group, sebuah organisasi yang telah mengembangkan model, teknologi, dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat desain berorientasi obyek dari IBM. UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi, yaitu:

1. *Use Case Diagram*

Use case diagram digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. *Use case diagram* terdiri atas diagram untuk *use case* dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi.

2. *Conceptual Diagram*

untuk memodelkan konsep-konsep yang ada di dalam aplikasi.

3. *Sequence Diagram*

Sequence diagram menjelaskan secara detil urutan proses yang dilakukan dalam sistem untuk mencapai tujuan dari *use case*, yaitu interaksi yang terjadi antar *class*, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

4. *Collaboration Diagram*

Collaboration diagram dipakai untuk memodelkan interaksi antar object di dalam sistem. Berbeda dengan *sequence diagram* yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration diagram* lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh obyek.

5. *State Diagram*

Untuk memodelkan perilaku obyek di dalam sistem.

6. *Activity Diagram*

Untuk memodelkan perilaku *Use Cases* dan obyek di dalam sistem.

7. *Class Diagram*

Class diagram merupakan diagram yang selalu ada pada permodelan sistem berorientasi obyek. *Class diagram* menunjukkan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan.

8. *Object Diagram*

untuk memodelkan struktur object.

9. *Component Diagram*

untuk memodelkan komponen object.

10. *Deployment Diagram*

untuk memodelkan distribusi aplikasi.

2.5.6 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*database management system*) atau DBMS yang *multithread*, *multi-user* dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. MySQL juga memiliki beberapa kelebihan, antara lain :

1. *Portability* :

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti windows, Linux, FreeBSD, Solaris dan lain-lain.
2. *Open Source* :

   MySQL didistribusikan secara open source (gratis), dibawah lisensi GPL sehingga dapat digunakan cuma-Cuma.
3. *Multi User* :

   MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance Tuning* :

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. *Coloumn Types* :

   MySQL memiliki tipe kolom yang sangat kompleks, seperti *integer*, *double*, *char*, *text*, *date* dll.
6. *Command and Function* :

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam *query*.

7. *Security* :

   MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta *password* terenkripsi.

8. *Scability and Limits* :

   MySQL mampu menangani *database* dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity* :

   MySQL dapat melakukan koneksi dengan clients menggunakan protokol TCP/IP, *Unix socket* (UNIX) atau *Named Pipes* (NT).

10. *Localisation* :

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface* :

    MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemograman dengan menggunakan fungsi API (*Aplication Programming Interface*).

12. *Clients and Tools* :

    MySQL dilengkapi dengan berbagai *tools* yang dapat digunakan untuk administrasi *database* dan pada setiap *tool* yang ada disertakan petunjuk *online*.

2.5.7 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.

PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti *JavaScript* yang diproses pada web browser (*client*).

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.