BAB III

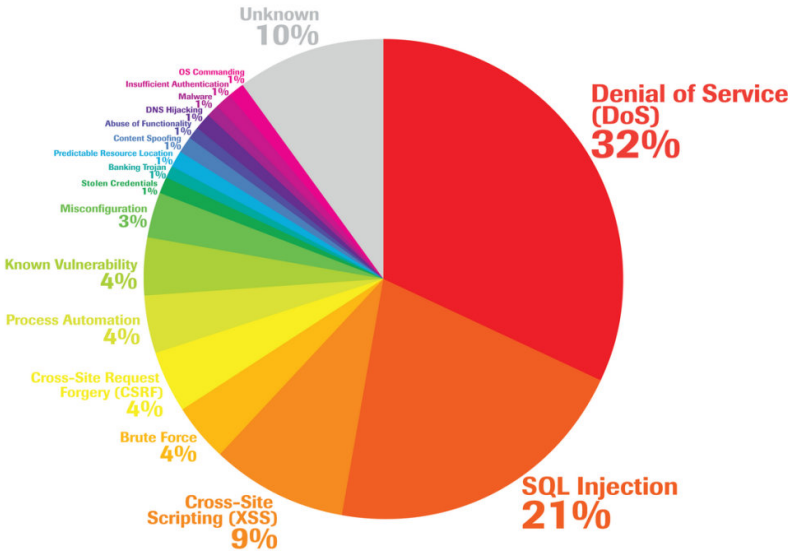
ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis Sistem adalah penguraian dari suatu masalah yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya. Bab ini akan menguraikan proses analisis sistem simulasi ancaman keamanan jaringan komputer pada aplikasi berbasis web.

3.1.1 Analisis Masalah

Berdasarkan data dari *Web Hacking Incidents Database* (WHID) pada tahun 2010, dari 75 jenis ancaman terhadap aplikasi web, sebanyak 21% merupakan ancaman *SQL Injection* dan 9% adalah ancaman *Cross Site Scripting*.[13]



Gambar 3.1 *Metode Ancaman yang Terjadi*

Ancaman *SQL Injection* merupakan ancaman yang terjadi pada database aplikasi web. Database digunakan sebagai media penyimpanan data yang berisi informasi berkaitan dengan aplikasi. Dalam penanganan ancaman *SQL Injection*, metode yang digunakan adalah *Base64_Encode,* namun metode ini masih rentan terhadap manipulasi *string*. Ancaman kedua adalah *Cross Site Scripting* yang terjadi pada tampilan aplikasi web. Salah satu penyebab ancaman ini adalah kurangnya metode *Input Encoding* dalam proses penyaringan karakter sehingga mengizinkan karakter tertentu dieksekusi. Kedua ancaman di atas menyulitkan para perancang aplikasi web untuk mengetahui mekanisme ancaman yang terjadi.

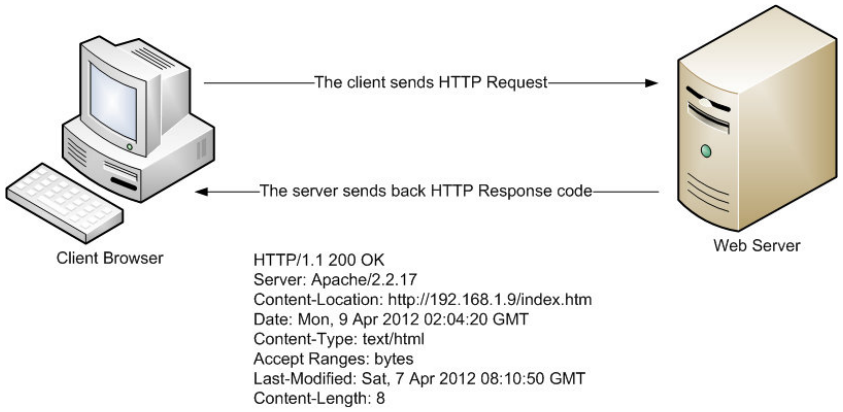
Berikut ini adalah hal-hal yang akan dianalisis, yaitu Analisis Protokol, Analisis *HTTP Request*, Analisis URL, Analisis Metode Ancaman, Analisis Metode Pertahanan dan Analisis Kebutuhan Infrastruktur.

3.1.2 Analisis Protokol

Protokol yang digunakan dalam penelitian ini adalah *Hyper Text Transfer Protocol* (HTTP). HTTP adalah protokol yang digunakan untuk mentransfer dokumen dalam *World Wide Web* (WWW), termasuk protokol ringan, tidak berstatus dan generik yang dapat digunakan pada berbagai macam dokumen dan protokol ini terdapat pada *port* 80.[10]

HTTP adalah sebuah protokol yang berfungsi untuk meminta/menjawab antara *client* dan *server*. Sebuah *client* HTTP seperti web browser, akan memulai permintaan dengan membuat hubungan TCP/IP ke *port* tertentu di *host* yang jauh. Sebuah *server* HTTP yang mendengarkan di *port* tersebut menunggu *client* mengirim kode permintaan (*request*), seperti "GET / HTTP/1.1" (yang akan

meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa *header* yang menjelaskan aspek dari permintaan tersebut, diikuti dengan badan dari data tertentu. Terdapat *header* yang bebas ditulis atau tidak, namun untuk *header* yang lain, seperti *host* diperlukan oleh protokol HTTP/1.1. Pada saat menerima *request* (dan pesan, bila ada), *server* akan mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan *error* karena terjadi kesalahan. Pada gambar 3.2 berikut ini, menggambarkan mekanisme pada Protokol HTTP.



Gambar 3.2 *Protokol HTTP*

3.1.2.1 HTTP Request

Terdapat dua buah tipe request HTTP yang paling umum digunakan yaitu GET dan POST yang juga dikenal sebagai *Request Method*. *Request* GET digunakan untuk mendapatkan atau menerima informasi dari *server*. Pada umumnya *request* GET digunakan untuk menerima *file* HTML atau *image*. Sedangkan *request* POST digunakan untuk mengirimkan data ke *server*. *Request* POST akan mengirimkan informasi *server* dalam bentuk HTML yang berisi data

yang dimasukkan oleh *client*. Kedua metode HTTP Requsest ini sering digunakan pada ancaman *Cross Site Scripting*.

Langkah awal dari *request* tersebut adalah dengan menentukan metode yang digunakan. Tabel 3.1 merincikan metode yang umum digunakan pada *HTTP Request*.

Tabel 3.1 *Metode HTTP/1.0*

Metode	Keterangan
GET	Mengambil informasi yang diminta dari sistem file. Jika file yang diminta merupakan file HTML statis, isi file akan ditampilkan. Tetapi jika file itu *server side scripting* misalnya PHP yang dinamis, maka web server akan memproses file PHP itu, mengeksekusi perintah-perintahnya, dan mengirim output perintah-perintah itu ke browser yang memintanya.
HEAD	Metode HEAD hampir sama dengan GET. Perbedaannya pada: HEAD tidak mengembalikan data yang diminta. Tetapi, kekuatan dari HEAD adalah ia akan merespon dengan meta-informasi seperti kode respon server, header tanggal, header server, dan sejenisnya.

	Karakteristik ini membuat penyerang bisa melihat satu per satu web server yang sedang menjalankan software web.
POST	Metode POST meminta kepada server agar menerima informasi yang diberikan dan menindaklanjutinya. Metode POST secara umum digunakan sewaktu PHP atau *scripting server side* diikutsertakan.

3.1.3 Analisis URL

URL merupakan sebuah mekanisme untuk mengenali sumber-sumber pada Web, SSL, atau server FTP, termasuk protokol *layer* aplikasi yang membuat *request* ke server web. Struktur umum URL adalah:[10]

```
protocol://server/path/to/resource?parameters
```

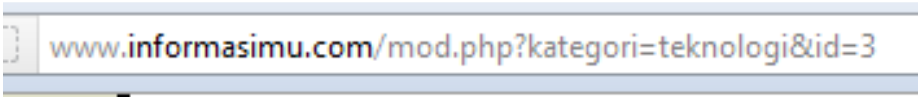
Tabel 3.2 *Komponen URL*

Komponen	Keterangan
Protocol	Protokol lapisan aplikasi. Kegunaan URL yang paling umum adalah meminta sumber-sumber dari web server (server HTTP). Karena itu protokol yang paling umum adalah

	http:. Protokol-protokol yang lain adalah https:, ftp:, ldap:, telnet:, pop3:, dan lain-lain, tergantung dukungan server dan browser apa yang digunakan.
Server	Nama DNS, nama Netbios, atau alamat IP dari sebuah *host* atau jaringan yang menjadi *host* sumber yang diminta.
Path/to/resource	Direktori *path*, termasuk nama sumber yang diminta. Sumber bisa dalam bentuk file statis atau sebuah aplikasi yang secara dinamis membangkitkan output.
Parameter	Secara opsional, parameter dapat dilewatkan ke sumber bila ia adalah sebuah aplikasi atau sebuah program yang secara dinamis membangkitkan output. Bagian URL yang mengkhususkan parameter disebut *Query String*.
?	Pemisah antara nama *resource* dengan parameter yang akan dilewatkan.

3.1.3.1 URL Page

URL Page adalah rangkaian karakter menurut suatu format standar tertentu, yang digunakan untuk menunjukkan alamat suatu sumber seperti dokumen dan gambar di Internet. Target pada simulasi ini beralamat di www.informasimu.com.



Gambar 3.3 *URL Page*

Analisis pertama bisa ditarik dari nama sumbernya, **mod.php**. Ekstensi **.php** menandakan bahwa file ini adalah file PHP yang merupakan *Opensource Language Programming*. File-file PHP berjalan secara khusus pada banyak Web Server Apache. Dengan demikian, www.informasimu.com kemungkinan besar adalah sebuah server yang menggunakan Linux pada Apache.

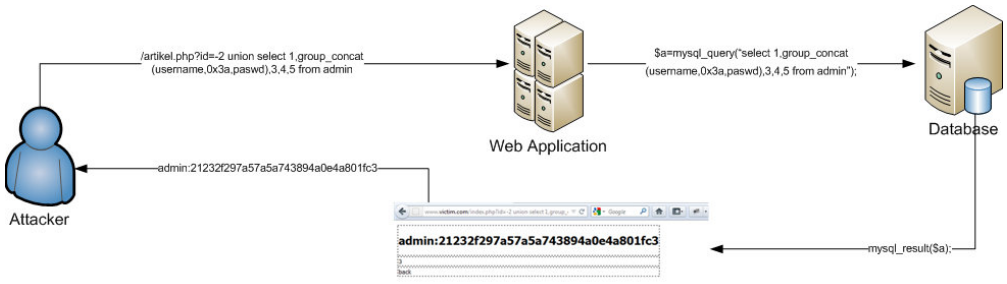
www.informasimu.com/mod.php?kategori=teknologi&id=3 , terlihat dari URL Page yang di *generate* ditemukan beberapa petunjuk. Parameter pertama yaitu **kategori=teknologi** menandakan bahwa berita yang sedang diakses adalah mengenai teknologi, dan berita tersebut pasti disimpan di database. Pilihan terpopuler dari platform database yang digunakan oleh produk opensource adalah MySQL. Parameter kedua yaitu **id=3**, menandakan bahwa berita dari kategori teknologi sedang mengakses id ke-3. Jadi file **mod.php** mempunyai mekanisme pengambilan data dari database kemudian menampilkan ke browser.

3.1.4 Analisis Metode Ancaman

Dalam penelitian ini, metode ancaman yang digunakan ada 2 macam yaitu *SQL Injection* dan *Cross Site Scripting.*

3.1.4.1 SQL Injection

SQL Injection adalah metode ancaman yang mengizinkan *client* untuk mengeksekusi database melalui URL dan mendapatkan akses untuk memperoleh informasi penting. Mekanisme ancaman dilakukan dengan memanfaatkan kesalahan pada kode program yang tidak di*filter*, sehingga menyebabkan terjadinya eksploitasi pada database.[2][3][8][9]



Gambar 3.4 *Mekanisme SQL Injection*

a. Mencari Celah Keamanan

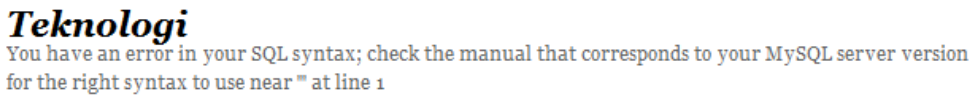
Ancaman SQL Injection dilakukan pada URL yang mempunyai id. Mekanisme ancaman dilakukan sebagai berikut.

http://www.informasimu.com/mod.php?kategori=teknologi&id=3

Tambahkan tanda petik satu (‘) di akhir id, sehingga menjadi:

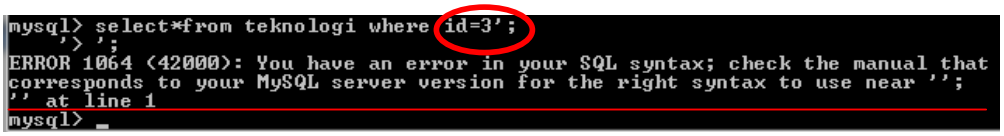
http://www.informasimu.com/mod.php?kategori=teknologi&id=3'

maka akan tampil pesan kesalahan seperti "*You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right...*", itu menadakan website mempunyai potensi celah keamanan *SQL Injection*.



Gambar 3.5 *Pesan Kesalahan Web Aplikasi*

Tanda petik satu(‘) pada akhir id digunakan untuk mengecek apakah variabel id difilter atau tidak. Apabila variabel tersebut tidak difilter maka akan menampilkan pesan kesalahan yang menyatakan kesalahan pada *query* SQL. Gambar berikut ini menjelaskan mengapa tanda petik satu(‘) pada akhir id dapat menghasilkan pesan kesalahan.



Gambar 3.6 *Pesan Kesalahan pada MySQL Console*

Dari pesan kesalahan tersebut, selanjutnya penyerang dapat menyimpulkan bahwa variabel pada id tidak di filter, sehingga dengan tidak adanya proses pem-filter-an pada id, semua karakter yang di*input*kan akan dijalankan menurut aturan query pada MySQL.

b. Menemukan Banyak Kolom

Setelah dapat menyimpulkan bahwa semua karakter yang di*input*kan akan dijalankan menurut aturan query pada MySQL, maka selanjutnya penyerang akan

menginputkan query tertentu. Untuk menemukan banyak kolom, digunakan statemen ORDER BY. Berikut adalah cara penggunaannya.

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 1

//tidak ada kesalahan

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 2

//tidak ada kesalahan

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 3

//tidak ada kesalahan

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 4

// tidak ada kesalahan

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 5

//tidak ada kesalahan

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 6

//tidak ada kesalahan

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 7

//tidak ada kesalahan

http://www.informasimu.com/mod.php?kategori=teknologi&id=3 order by 8

// ada kesalahan

pesan kesalahan seperti “Unknown column '8' in 'order clause'” yang menyatakan bahwa tabel tersebut tidak menemukan tersebut kolom kedelapan.

c. Menemukan Informasi Database

Untuk menemukan informasi database, gunakan fungsi *UNION*. *UNION* adalah fungsi yang digunakan untuk menggabungkan lebih dari satu *query* dalam satu perintah SQL. Penggunaan fungsi ini adalah sebagai berikut.

http://www.informasimu.com/mod.php?kategori=teknologi&id=-3 union select 1,version(),3,4,5,6,7. Hasil dari *query* tesebut menampilkan beberapa angka dan versi dari database yang digunakan.



Gambar 3.7 *Web Aplikasi Menampilkan Angka*

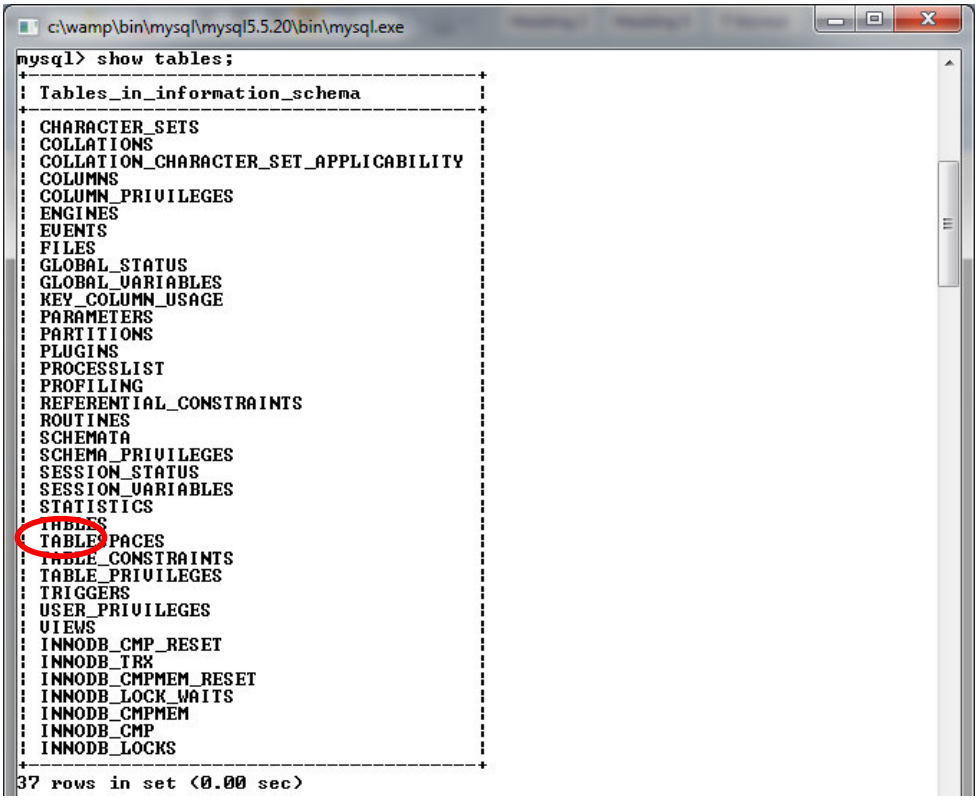
3.1.4.2 Analisis Information_schema

Pada gambar 3.7 web aplikasi menampilkan versi database. Versi database yang digunakan adalah versi 5. Pada database versi 5, terdapat database *default* yang dibuat ketika MySQL di *install*. Database tersebut adalah *information_schema*. Gambar 3.8 berikut adalah database *information_schema* pada MySQL.[3]



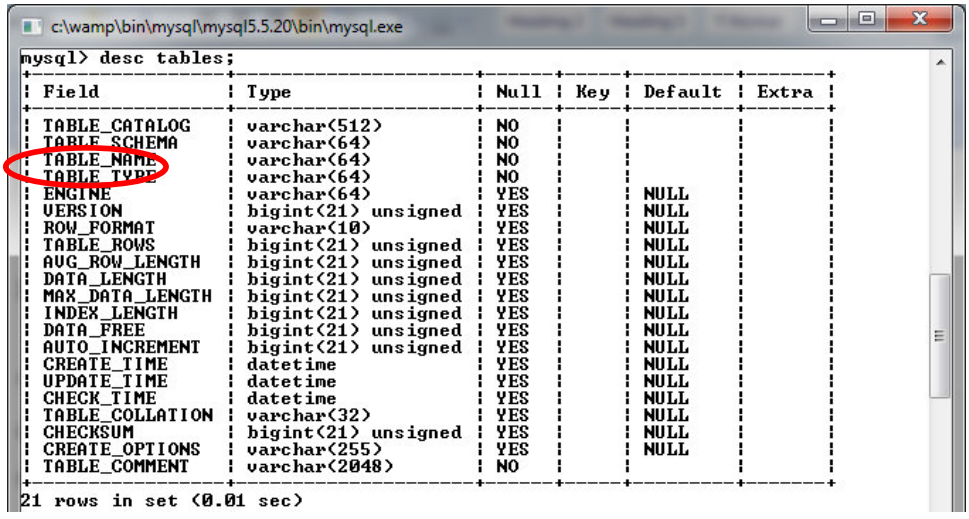
Gambar 3.8 *Database information_schema*

Database ini menyimpan seluruh informasi tabel dan kolom yang ada pada database MySQL. Gambar 3.9 berikut merupakan tabel dari database *information_schema.*



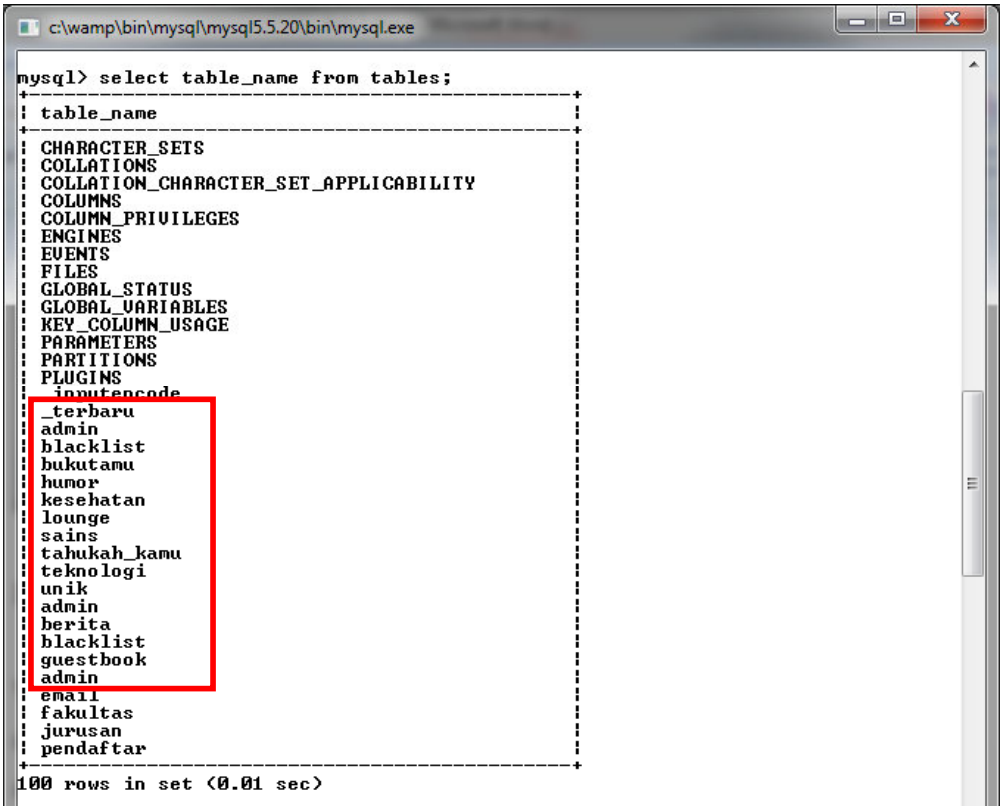
Gambar 3.9 *Tabel pada Information_schema*

Tabel *TABLES* pada *information_schema* menyimpan informasi tabel-tabel yang terdapat dalam database. Gambar 3.10 berikut menampilkan deskripsi dari *TABLES*.



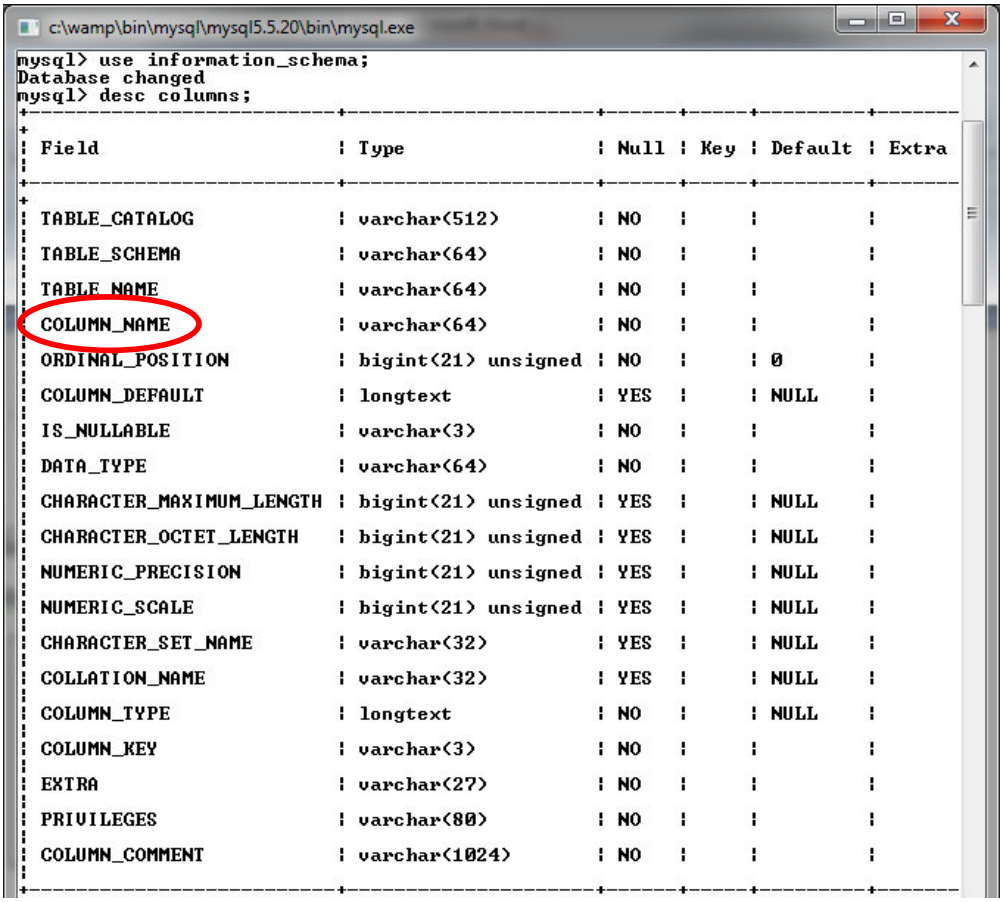
Gambar 3.10 *Deskripsi TABLES*

Di dalam *TABLES* terdapat *field* yang bernama *table_name*. Tabel tersebut berisi informasi dari semua tabel yang ada di database. Gambar 3.11 menampilkan isi dari *table_name*.



Gambar 3.11 *Isi table_name*

Gambar 3.11 merupakan isi dari *table_name*. Sebagian dari gambar tersebut tidak ditampilkan karena tabel yang ada pada *table_name* terlalu banyak. Selain dari tabel, *information_schema* juga menyimpan informasi kolom yang terdapat pada tabel. Gambar 3.12 menampilkan deskripsi *column* pada database *information*_schema.



Gambar 3.12 *Deskripsi COLUMNS*

Di dalam *COLUMNS* terdapat *field* yang bernama *column_name*. *Field* tersebut berisi informasi dari semua kolom yang ada di database. Gambar 3.13 menampilkan isi dari *column_name*.



Gambar 3.13 *Isi column_name*

Gambar 3.13 merupakan isi dari *column_name*. Sebagian dari gambar tersebut tidak ditampilkan karena kolom yang ada pada *column_name* terlalu banyak.

Database *information_schema* akan digunakan pada saat ancaman *SQL Injection* dijalankan.

3.1.4.3 Penyebab SQL Injection

SQL Injection terjadi karena ada suatu variabel yang tidak di filter. Variabel tersebut berfungsi sebagai penampung id yang akan me-*lookup* ke database. Apabila variabel ini tidak di filter, maka semua jenis tipe data akan dieksekusi melalui variabel ini. Berikut ini adalah kode yang menyebabkan ancaman *SQL Injection* terjadi.[2][3]

```
43. $id=@$_GET['id'];
44. if($id) {
45.     echo "<table width=\"50%\">";
46.     $sql="select*from $kategori where id=$id";
47.     $berita=mysql_query($sql)or die (mysql_error());
```

Gambar 3.14 *Kode yang Menyebabkan SQL Injection*

Pada gambar 3.14 merupakan bagian kode dari file **mod.php**. File **mod.php** adalah sebuah file yang berfungsi untuk me-*lookup* berita dari database kemudian menampilkannya ke browser. Ancaman *SQL Injection* terjadi karena pada variabel **id** tidak di *filter*. Variabel ini berada pada *line* 43.

SQL Injection bekerja pada variabel id yang tidak di *filter*. Pada variabel id ini dapat disisipkan kode yang dapat memanipulasi *request* ke database. Berikut ini adalah salah satu penggunaan kode yang telah disisipkan pada sebuah URL:

`http://www.informasimu.com/mod.php?kategori=teknologi&id=-3 union select 1,version(),3,4,5,6,7`

ketika URL tersebut dieksekusi, maka yang terjadi pada kode adalah sebagai berikut:

```
43.  $id=@$_GET['id'];
44.  if($id) {
45.      echo “<table width=\”50%\”>”;
46.      $sql=”select*from  teknologi  where  id=-3  union
         select 1,version(),3,4,5,6,7”;
47.      $berita=mysql_query($sql)or die (mysql_error());
```

Gambar 3.15 *Eksekusi Query SQL Injection*

Pada gambar 3.15 di *line* 46, *query* berhasil dimanipulasi oleh *attacker* sehingga *query* tersebut tidak hanya meminta data dari kategori **teknologi** dengan id **ke-3**, namun juga menampilkan versi dari database yang digunakan. Fungsi version() merupakan fungsi yang terdapat pada MySQL untuk mengetahui versi database yang digunakan. Dengan berjalannya *query* ini, maka *attacker* dapat menjalankan *query* lain dengan fungsi-fungsi tertentu yang dapat membahayakan database hingga webserver.

Salah satu metode yang digunakan untuk mengurangi ancaman *SQL Injection* adalah metode *Base64_Encode*. Namun metode ini masih memiliki kekurangan. Berikut ini adalah analisis dari Metode *Base64_Encode*.

3.1.4.3.1 Metode Base64_Encode

Base64 Encoding adalah skema pengkodean yang mengkodekan data biner dan menerjemahkannya ke dalam representasi basis 64. Skema *encoding* base64 digunakan ketika ada kebutuhan untuk menyandikan data biner yang perlu disimpan dan ditransfer melalui media yang dirancang untuk menangani data tekstual. Tabel 3.3 merupakan karakteristik dari base64.[11]

Tabel 3.3 *Karakteristik Base64*

Value	*Char*		*Value*	*Char*		*Value*	*Char*		*Value*	*Char*
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	=

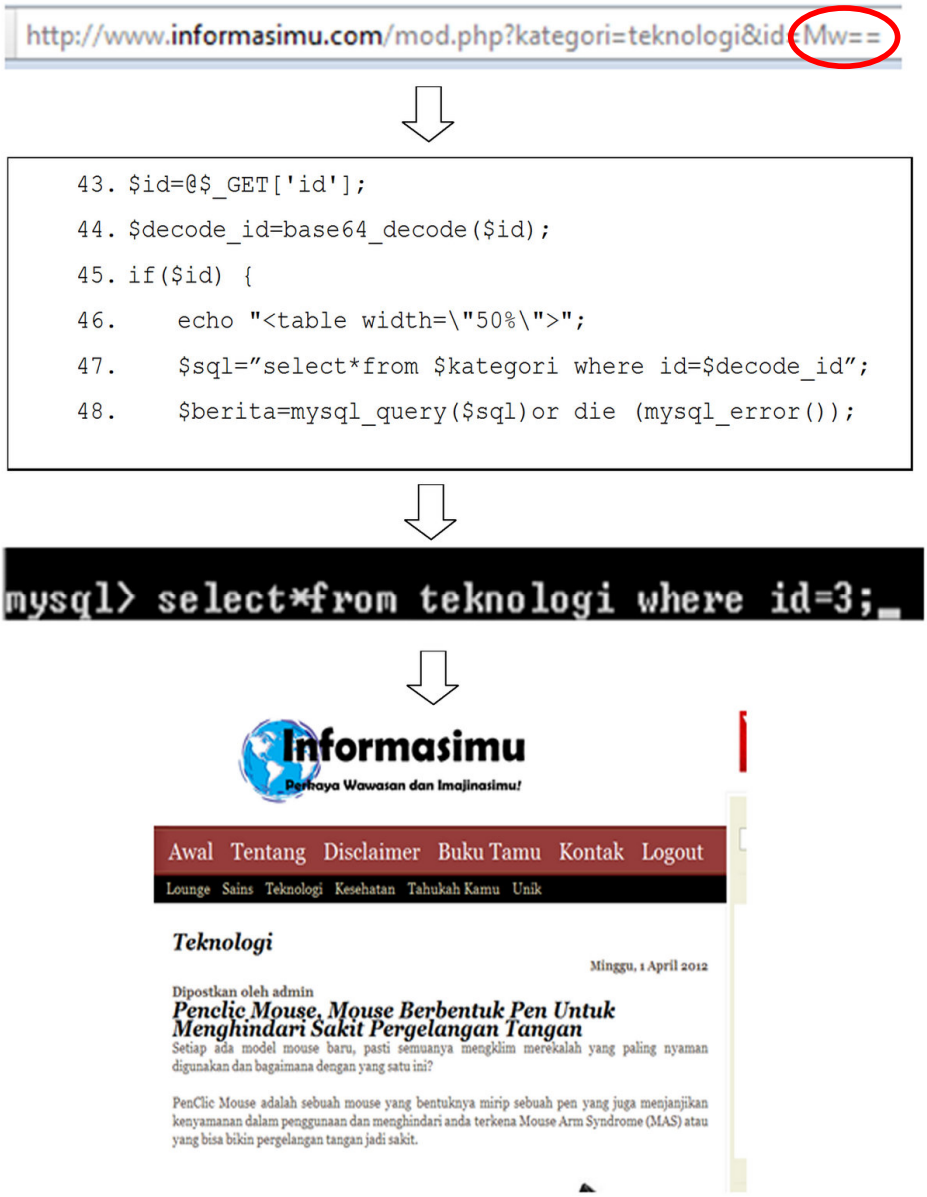
Salah satu metode dalam mengurangi ancaman *SQL Injection* adalah dengan menggunakan *Base64_Encode*. Metode ini menyamarkan id yang membawa berita dari database. Id yang disamarkan melalui metode ini cukup dapat mencegah ancaman, namun jenis samaran ini masih dapat menyebabkan *SQL Injection* terjadi, karena id hanya di *encode* dengan base64, sehingga *attacker*

dapat menjalankan *SQL Injection* hanya dengan mengganti karakter menjadi *base64*. Penggunaan base64_metode dapat dilihat pada URL berikut:



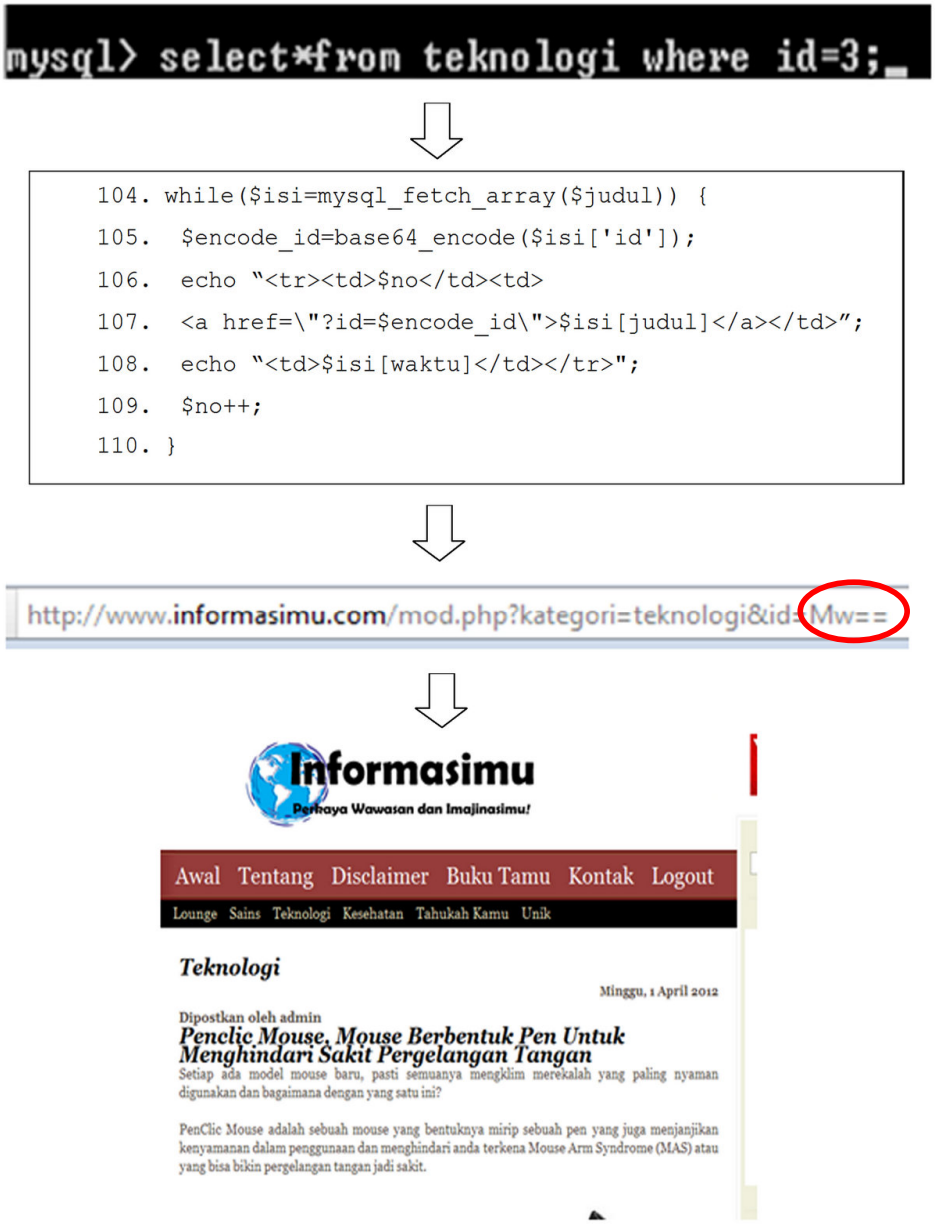
Gambar 3.16 *ID yang Berbentuk Base64*

Mw== merupakan representasi dari angka 3 yang telah diubah menjadi base64_encode. Representasi ini memiliki mekanisme yang sama dalam me-*lookup* berita ke database. Di dalam kode, id yang berupa angka akan di-*encode* terlebih dahulu menggunakan base64, kemudian pada saat melakukan *query* ke database, id akan di-*decode* kembali. Berikut ini adalah kode yang menunjukkan proses *decode* dan *encode* terhadap variabel id:



Gambar 3.17 *Proses Decode Pada ID*

Pada gambar 3.17 di *line* 44 adalah proses *decode* terhadap id yang telah di *encode* sebelumnya. Berikut ini adalah proses *encode* terhadap variabel id.



Gambar 3.18 *Proses Encode Pada ID*

Pada gambar 3.17 dan 3.18 merupakan bagian kode dari file **mod.php**. Proses *encode* pada gambar 3.18 yaitu kode akan mengecek apakah variabel id telah terisi oleh sebuah nilai, apabila variabel tersebut tidak terisi nilai maka id akan di *encode*, namun apabila variabel tersebut terisi nilai yang telah ter-*encode*, maka

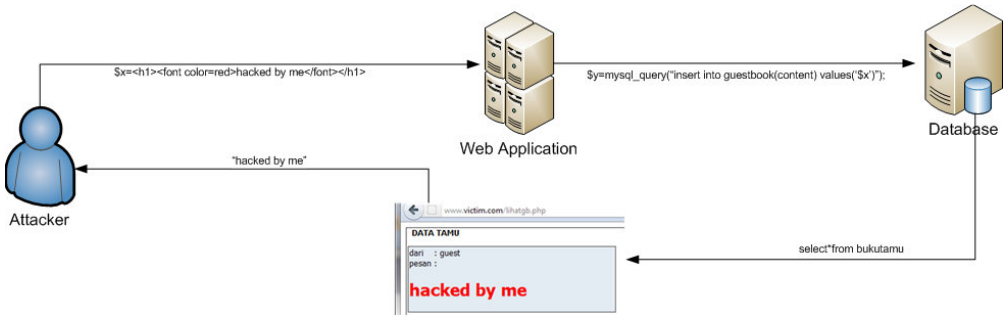
proses selanjutnya merupakan proses *decode* terhadap nilai variabel id, kemudian melakukan *query* ke database. *Line* 105 merupakan proses *encode* terhadap id.

Proses penyamaran id ini sekilas memang tidak diketahui oleh pengguna. Namun *attacker* dengan mudah untuk memanipulasi variabel ini sehingga dapat menjalankan *query* yang berbahaya bagi database dan server.

Secara logika, **id=Mw==** yang terdapat pada URL merupakan representasi dari **id=3** . Apabila dianalisis lebih lanjut, variabel **id** hanya akan membaca nilai yang berbentuk base64 saja, sehingga nilai apapun selain base64 tidak akan diproses oleh *query*. Melalui cara ini, *attacker* dapat melakukan eksekusi *query* hanya dengan mengkonversikan nilai menjadi bentuk base64.

3.1.4.4 Cross Site Scripting

Cross Site Scripting adalah metode ancaman yang memaksa situs web untuk menampilkan kode berbahaya, yang kemudian dijalankan pada browser web pengguna. Mekanisme ancaman dilakukan dengan memanfaatkan kesalahan pada kode program yang tidak di *filter*. Kode tersebut akan dimasukkan ke dalam *form*, di antaranya *form* buku tamu atau *form* pencarian.[3][4]



Gambar 3.19 *Mekanisme Cross Site Scripting*

3.1.4.5 Penyebab Cross Site Scripting

Cross Site Scripting terjadi karena tidak adanya penangan nilai masukan pada variabel sehingga masukan tersebut dieksekusi begitu saja sesuai nilai yang ada. Apabila variabel tersebut tidak di filter, maka semua nilai masukan akan dieksekusi. Berikut ini adalah kode yang menyebabkan ancaman *Cross Site Scripting* terjadi.

```
10. $nama=$_POST['nama'];
11. $alamat=$_POST['alamat'];
12. $pesan=$_POST['pesan'];
13. $kode=$_POST['kode'];
14. $pattern=$_POST['asd'];
15.
16. if(($nama=="")||($alamat=="")||($pesan=="")||
    ($kode!=$pattern)) {
17. echo "<p><b>semua field harus diisi!</b></p>";
18.  }
19. else
20. if(($nama!="")&&($alamat!="")&&($pesan!="")&&
   ($kode==$pattern)) {
21.
22. $masuk=mysql_query("insert into bukutamu values
    (NULL,'$nama','$alamat','$pesan')");
23.  if($masuk) {
24.      echo "<p><b>pesan berhasil disimpan!</b></p>";
25.      }
26. else echo "<p><b>pesan gagal disimpan</b></p>";
```

Gambar 3.20 *Kode yang Menyebabkan Cross Site Scripting*

Pada gambar 3.20 merupakan bagian kode dari file **contoh.php**. File **contoh.php** adalah sebuah file yang berfungsi untuk memproses masukan, sehingga masukan disimpan ke database. Ancaman *Cross Site Scripting* terjadi

karena pada variabel **nama, alamat,** dan **pesan** tidak di *filter*. Variabel ini berada pada *line* 10-12.

Pada variabel tersebut dapat disisipkan kode yang dapat memanipulasi halaman website sehingga halaman tersebut menjadi tidak teratur bahkan kode yang disisipkan dapat mencuri *cookie* dari pengguna yang dapat menyebabkan terjadinya eksploitasi pada akun pengguna. Berikut ini adalah salah satu penggunaan kode yang telah disisipkan pada sebuah *form* buku tamu[14]:



Gambar 3.21 *Script yang disisipkan pada Buku Tamu*

Setelah tombol *submit* di tekan, maka akan keluar gambar berikut:



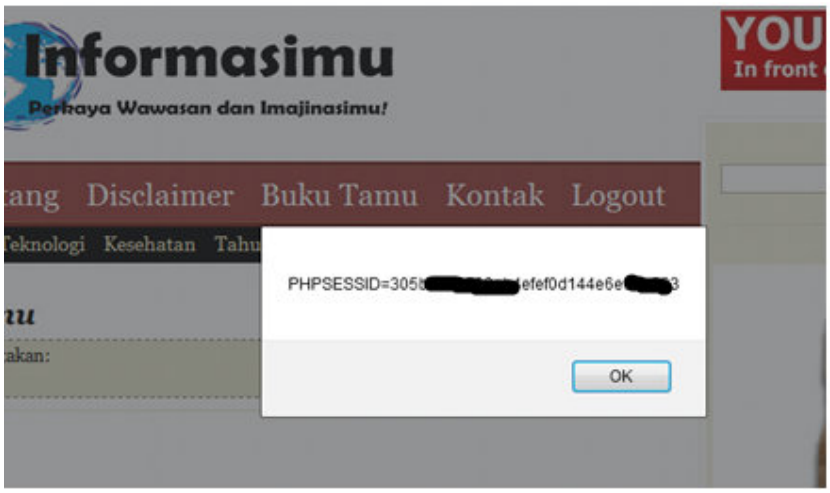
Gambar 3.22 *Script yang dijalankan pada Buku Tamu*

Gambar 3.23 berikut ini adalah penyisipan *script* yang akan menampilkan *cookie* pengguna.



Gambar 3.23 *Script cookie yang disisipkan*

Gambar 3.24 berikut ini adalah eksekusi kode yang dapat menampilkan *cookie* pengguna.



Gambar 3.24 *Script cookie yang dijalankan pada Buku Tamu*

Ketika nilai tersebut di-*submit*, maka contoh.php akan mengeksekusi kode yang telah disisipkan untuk disimpan ke database. Berikut ini adalah kode dari contoh.php yang telah disisipkan oleh kode pada gambar 3.21:

```
10. $nama=”Guest”;
11. $alamat=”d1gdo.com”;
12. $pesan=”<font color=red><h1>halo admin</h1></font>”;
13. $kode=”164184”;
14. $pattern=”164184”;
15.
16. if(($nama=="")||($alamat=="")||($pesan=="")||
    ($kode!=$pattern)) {
17. echo "<p><b>semua field harus diisi!</b></p>";
18.  }
19. else
20. if(($nama!="")&&($alamat!="")&&($pesan!="")&&
   ($kode==$pattern)) {
21.
22. $masuk=mysql_query("insert into bukutamu values
    (NULL,'$nama','$alamat','$pesan')");
23.  if($masuk) {
24.      echo "<p><b>pesan berhasil disimpan!</b></p>";
25.      }
26. else echo "<p><b>pesan gagal disimpan</b></p>";
```

Gambar 3.25 *Eksekusi Kode Cross Site Scripting*

Pada gambar 3.25 di *line* 10-12, kode dimasukkan oleh *attacker* sehingga contoh.php menyimpan kode tersebut ke database layaknya sebuah *inputan* biasa. Dengan dijalankannya kode ini, maka *attacker* dapat menyisipkan kode lain dengan fungsi-fungsi tertentu yang dapat membahayakan website hingga pengguna aplikasi.

Dalam penangan ancaman *Cross Site Scripting*, salah satu metode yang digunakan adalah *Input Encoding*. Namun, metode ini masih mempunyai kekurangan apabila tidak disertakan dengan metode lain. Berikut ini adalah analisis dari metode *Input Encoding*.

3.1.4.6 Metode Input Encoding

Salah satu metode dalam mengurangi ancaman *Cross Site Scripting* adalah dengan menggunakan *Input Encoding.* Metode ini meng-*encode* masukan tertentu yang dianggap bahaya menjadi masukan yang tidak berbahaya bagi sistem. Penggunaan metode ini dengan memilih satu per satu karakter yang dianggap bahaya kemudian menggantinya dengan karakter biasa. Kekurangan dalam metode ini adalah bentuk karakter yang berbahaya sangat banyak, dan tidak semua *web developer* mengetahui karakter tersebut. Penggunaan metode *Input Encoding* dapat dilihat pada gambar berikut:[9]

Tabel 3.4 *Blacklist Karakter*

Id	Input	Output
1	<script>	<skrip>
2	alert	peringatan
3	‘	.
4	}	a
5	{	b
6	=	c
7	&	d

Apabila terdapat masukan berupa “script” maka sistem otomatis akan mengganti masukan tersebut menjadi “skrip”, atau apabila terdapat masukan berupa “=” maka sistem otomatis akan mengganti masukan tersebut menjadi “c”, begitu seterusnya. Kelemahan dari metode ini adalah *web developer* harus mendefinisikan satu per satu karakter/kata yang berpotensi dapat membahayakan

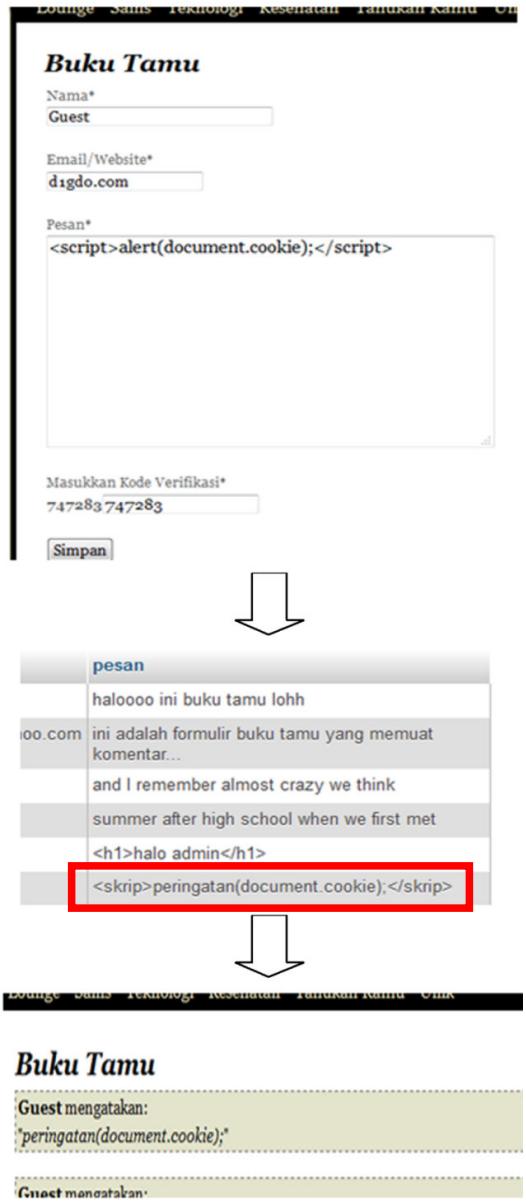
sistem, selain itu tidak semua *web developer* mengetahui bentuk karakter yang dapat membahayakan sistem yang dibuat.

Gambar 3.26 berikut ini adalah fungsi *blacklist* yang digunakan untuk menggantikan karakter yang dianggap bahaya bagi sistem.

```
<?php
function blacklist($komen) {
$pilih=mysql_query("select * from blacklist");
 while($ambil=mysql_fetch_array($pilih)) {
      $komen=str_ireplace($ambil[1],$ambil[2],$komen);
}     return $komen;
}
?>
```

Gambar 3.26 *Fungsi Blacklist Karakter*

Gambar 3.27 berikut ini merupakan proses penggantian karakter.



Gambar 3.27 *Proses Penggantian Karakter*

3.1.5 Analisis Metode Pertahanan

Validating Input adalah proses pengujian masukan yang diterima oleh aplikasi terhadap standar yang telah didefinisikan. Ada dua jenis pendekatan yang bisa digunakan yaitu *Whitelist Validation* dan *Blacklist Validation*.

3.1.5.1 Whitelist Validation

Whitelist Validation biasa disebut juga dengan *Positive Validation* adalah metode yang hanya menerima masukan yang tidak berbahaya bagi sistem. Hal ini dapat melibatkan validasi sesuai dengan panjang, jenis yang diharapkan atau ukuran, rentang numerik, atau standar format lain sebelum menerima masukan untuk diproses lebih lanjut. Berikut ini merupakan pertimbangan untuk penggunaan metode *Whitelist Validation*:[2][4][9]

1. *Data Type*

   Tipe data dalam *Whitelist Validation* harus sesuai dengan yang diharapkan. *Data Type* digunakan untuk mengecek jenis data. Apakah jenis data sudah benar atau tidak, jika nilai yang seharusnya numerik, apakah numerik atau tidak. Berikut ini adalah fungsi yang dibuat untuk *Data Type*.

```
//fungsi pengecekan Data Type
function checkint($number) {
      $check=(int)$number;
      return $check;
}
```

Gambar 3.28 *Fungsi Data Type*

2. *Data Size*

Data Size digunakan untuk mengecek ukuran data. Jika data adalah string, apakah panjang string tersebut sesuai atau tidak. Apabila berupa angka, apakah angka tersebut terlalu besar dari ukuran yang telah ditentukan atau tidak. Berikut ini adalah fungsi yang dibuat untuk *Data Size.*

```
//fungsi pengecekan Data Size
if(!function_exists("panjang")) {
  function panjang($char) {
    $checkz = null;
    $tes=strlen($char);
     if($tes>4)
     $checkz=header('location:./');
  return $checkz;
  }
}
```

Gambar 3.29 *Fungsi Data Size*

3. *Data Range*

Data Range digunakan untuk mengecek rentang data. Rentang data tidak boleh melebihi dari batas rentang yang telah ditentukan. Berikut ini adalah fungsi yang dibuat untuk *Data Range.*

```
//fungsi pengecekan Data Range
function checkrange($number) {
      $checkr=range(1,999999);
      return $checkr;
}
```

Gambar 3.30 *Fungsi Data Range*

4. *Data Content*

   Isi data harus ditentukan. Apakah data yang dimasukkan sesuai dengan aturan yang telah ditentukan atau tidak. Misalnya, apakah data tersebut memenuhi ketentuan dari sebuah kode pos, atau apakah data itu hanya berisi karakter biasa saja. Metode umum dalam melakukan validasi konten adalah dengan menggunakan *Regular Expression* (RegEx). Berikut ini adalah fungsi yang dibuat untuk *Data Content.*

```
//fungsi preg_match
function cek_params($id)
{
if(preg_match('/\s/', $id))
      exit('attack'); // no whitespaces
if(preg_match('/[\'"]/', $id))
      exit('attack'); // no quotes
if(preg_match('/[\/\\\\]/', $id))
      exit('attack'); // no slashes
}
```

Gambar 3.31 *Fungsi Data Content*

3.1.5.2 Blacklist Validation

Blacklist Validation biasa disebut juga dengan *Negative Validation* adalah metode yang menolak masukan yang berbahaya bagi sistem. Pada umumnya metode ini melibatkan masukan yang mengandung karakter, atau pola yang dianggap bisa membahayakan sistem.

Pendekatan ini umumnya lebih lemah dari *Whitelist Validation* karena karakter atau pola yang berpotensi membahayakan sistem sangat banyak, dan karakter tersebut kemungkinan akan selalu bertambah, tidak lengkap, dan sulit untuk selalu diperbaharui.

Metode umum dalam menerapkan *Blacklist Validation* juga menggunakan *Regular Expression*, yang menyimpan karakter yang dapat membahayakan sistem, seperti contoh berikut:

'|%|--|;|/*|*|_|\[|@|xp_

Secara umum, *Whitelist Validation* mempunyai kelebihan dari *Blacklist Validation*. Ketika ada set karakter besar yang akan digunakan dalam aplikasi, misalnya bahasa Cina dan Jepang, maka metode *Blacklist Validation* akan sulit diterapkan karena terdapat input data yang kompleks, dan input data tersebut tidak dapat ditentukan dengan mudah. Dalam kasus ini, *Whitelist Validation* merupakan metode yang paling tepat untuk digunakan. Selain itu tambahkan fungsi pengontrol lain seperti pengkodean output untuk memastikan bahwa informasi yang akan dikirimkan ke database dapat dilakukan dengan benar.

3.1.5.3 Validating Input PHP

Dalam kode PHP, memiliki beberapa fungsi yang dapat digunakan sebagai validasi terhadap inputan. Beberapa fungsi tersebut adalah:

preg_match (regex, $string) Mencocokkan pola inputan menggunakan *Regular Expression* terhadap variabel **$string**.

```
//fungsi preg_match
function cek_params($id)
{
if(preg_match('/\s/', $id))
      exit('attack'); // no whitespaces
if(preg_match('/[\'"]/', $id))
      exit('attack'); // no quotes
if(preg_match('/[\/\\\\]/', $id))
      exit('attack'); // no slashes
}
```

Gambar 3.32 *Fungsi preg_match*

is_ <jenis> (input) Memeriksa apakah input adalah sesuai dengan jenisnya. Misalnya, is_numeric ().

```
//fungsi pengecekan Data Size
if(!function_exists("panjang")) {
  function panjang($char) {
    $checkz = null;
    $tes=strlen($char);
     if($tes>4)
     $checkz=header('location:./');
  return $checkz;
  }
}
```

Gambar 3.33 *Fungsi is_numeric*

strlen (input) Memeriksa panjang input.

```
//fungsi pengecekan Data Type
function checkpanjang($string) {
      $check=strlen($string);
       if($check>4) {
         header('location:./');
      }
      return $check;
}
```

Gambar 3.34 *Fungsi strlen*

htmlentities() Mengkonversi karakter-karakter tertentu dalam tag HTML, hasil fungsi ini akan menerjemahkan tag-tag HTML sebagai teks biasa.

```
//fungsi pengecekan Data Type
function checkhtml($string) {
      $check=htmlentities($string);
      return $check;
}
```

Gambar 3.35 *Fungsi htmlentities*

strip_tags() Menghilangkan tag-tag HTML dan PHP dalam sebuah string.

```
//fungsi pengecekan Data Type
function checktag($string) {
      $check=strip_tags($string);
      return $check;
}
```

Gambar 3.36 *Fungsi strip_tags*

trim() Menghapus spasi di kanan dan kiri teks.

```
//fungsi pengecekan Data Type
function checkspasi($string) {
      $check=trim($string);
      return $check;
}
```

Gambar 3.37 *Fungsi trim*

3.1.6 Kebutuhan Pendukung Infrastruktur

Kebutuhan akan infrastruktur terbagi menjadi dua macam, yaitu perangkat keras dan perangkat lunak, yang keduanya saling mendukung satu sama lain.

3.1.6.1 Kebutuhan Perangkat Keras

Kebutuhan hardware terdiri dari beberapa perangkat keras meliputi PC sebagai server, PC sebagai *attacker*, dan modem.

a. PC Server

Komputer server menyediakan layanan-layanan yang akan diakses oleh *client*, seperti *web server* dan *file server*. Adapun spesifikasinya adalah sebagai berikut:

Tabel 3.5 *Spesifikasi Perangkat Keras Server*

No	Komponen	Keterangan
1	Processor	Intel(R) Xeon(R) CPU E31245 @ 3.30GHz
2	Memori	1GB
3	Harddisk	40GB
4	Operating System	CentOS 5

b. PC Attacker

Komputer *attacker* digunakan untuk menyerang aplikasi web yang terdapat pada komputer server.

Tabel 3.6 *Spesifikasi Perangkat Keras Attacker*

No	Komponen	Keterangan
1	Processor	Intel Core i3
2	Memori	3GB
3	Harddisk	500GB
4	Operating System	Backtrack 5 / Windows 7

3.1.6.2 Kebutuhan Perangkat Lunak

Software yang digunakan meliputi Kloxo, SSH (Secure Shell), VNC (Virtual Network Computing), dan FileZilla.

a. Kloxo

Kloxo adalah sebuah kontrol panel ringan berbasis opensource. Kloxo memiliki semua fitur yang disertakan dalam semua panel kontrol terkemuka seperti mail, spam filter, PHP, CGI, Perl, suexec, FTP, MySQL dan lain-lain.

b. SSH

Secure Shell atau SSH adalah protokol jaringan yang memungkinkan pertukaran data melalui saluran aman antara dua perangkat jaringan. Terutama banyak digunakan pada sistem berbasis Linux dan Unix untuk mengakses akun shell, SSH dirancang sebagai pengganti Telnet dan shell remote tak aman lainnya, yang mengirim informasi, terutama

kata sandi, dalam bentuk teks sederhana yang membuatnya mudah untuk dicegat. Enkripsi yang digunakan oleh SSH menyediakan kerahasiaan dan integritas data melalui jaringan yang tidak aman seperti Internet.

SSH menggunakan kriptografi kunci publik untuk mengotentikasi komputer *remote* dan biarkan komputer *remote* untuk mengotentikasi pengguna, jika perlu. SSH biasanya digunakan untuk login ke mesin *remote* dan mengeksekusi berbagai perintah, tetapi juga mendukung *tunneling*, *forwarding TCP port* dan X11 connections; itu dapat mentransfer file menggunakan terkait SFTP atau SCP protocols. SSH menggunakan klien-server model. Yang standar TCP port 22 telah ditetapkan untuk menghubungi server SSH. Sebuah klien program SSH ini biasanya digunakan untuk membangun koneksi ke SSH daemon untuk dapat diremote. Keduanya biasanya terdapat pada sistem operasi modern, termasuk Mac OS X, Linux, FreeBSD, Solaris dan OpenVMS. Tersedia versi berpemilik, freeware dan open source untuk berbagai tingkat kerumitan dan kelengkapan.

c. VNC

VNC adalah sebuah *tool remote control*. *Tool* ini digunakan untuk melakukan remote desktop yang biasa digunakan pada platform Linux atau Windows. Melaui VNC seseorang dapat mengontrol server atau komputer dari jarak jauh.

d. FileZilla

FileZilla atau juga dikenal dengan sebutan FileZilla Client, adalah salah satu software *File Transfer Protocol* (FTP) gratis, open source, *cross-platform*. Software ini mendukung FTP, SFTP, dan FTPS (FTP di SSL/TLS). Fitur utama dari Filezilla adalah :

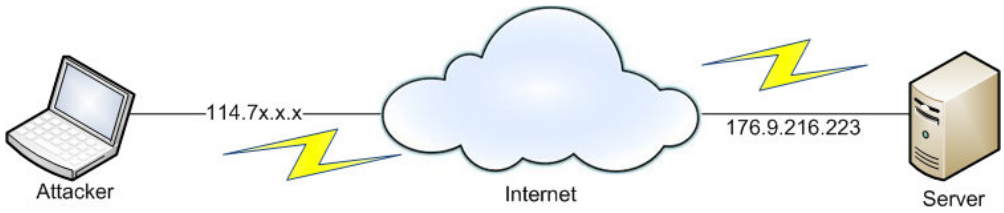
1) Site manager (Manajer situs). Mengizinkan pengguna untuk membuat daftar situs FTP beserta data koneksinya, seperti nomor port yang akan digunakan, protokol yang digunakan, dan apakah akan menggunakan log anonim atau normal. Untuk log normal, nama pengguna dan kata sandinya tersebut akan disimpan. Penyimpanan kata sandi adalah opsional.
2) Message log (Log pesan). Ditampilkan di bagian atas jendela. Fitur ini menampilkan output berjenis konsol (console-type) yang menunjukkan perintah yang dikirim oleh FileZilla dan respon yang diterima dari server.
3) File and folder view. Ditampilkan di bawah pesan log (Message log), menyediakan sebuah tampilan grafis antarmuka untuk FTP. Pengguna dapat menavigasi folder dan melihat dan mengubah isinya pada komputer lokal dan server dengan menggunakan tampilan antarmuka gaya Explorer. Pengguna dapat men-drag dan drop file antara komputer lokal dan server.
4) Transfer queue (Transfer antrian). Ditampilkan di sepanjang bagian bawah jendela, menunjukkan status real-time setiap antrian atau transfer file yang aktif.

3.2 Perancangan Sistem

Pada perancangan sistem akan dibahas mengenai perancangan sistem simulasi ancaman, yang meliputi perancangan arsitektur simulasi terhadap aplikasi web.

3.2.1 Arsitektur Simulasi

Perancangan jaringan yang akan digunakan terdiri dari sebuah komputer sebagai server, dan komputer *attacker* dengan bentuk topologi seperti yang ditunjukan pada gambar 3.38:



Gambar 3.38 *Konfigurasi Jaringan pada Simulasi*

Komputer server yang berfungsi sebagai infrastruktur komunikasi yang mempunyai mekanisme protokol komunikasi data HTTP dengan *IP Address* 176.9.216.223. Pada komputer ini menyediakan aplikasi web berupa portal informasi.

Pada komputer *attacker* terkoneksi dengan internet menggunakan modem HSDPA dengan kecepatan akses maksimal 256 KBps dan mendapatkan IP Publik dinamis (114.7x.x.x).

Skenario yang akan dilakukan yaitu *attacker* yang terkoneksi dengan internet akan mengakses layanan pada server yaitu sebuah portal informasi, kemudian *attacker* menginjeksikan kode tertentu pada aplikasi web menggunakan

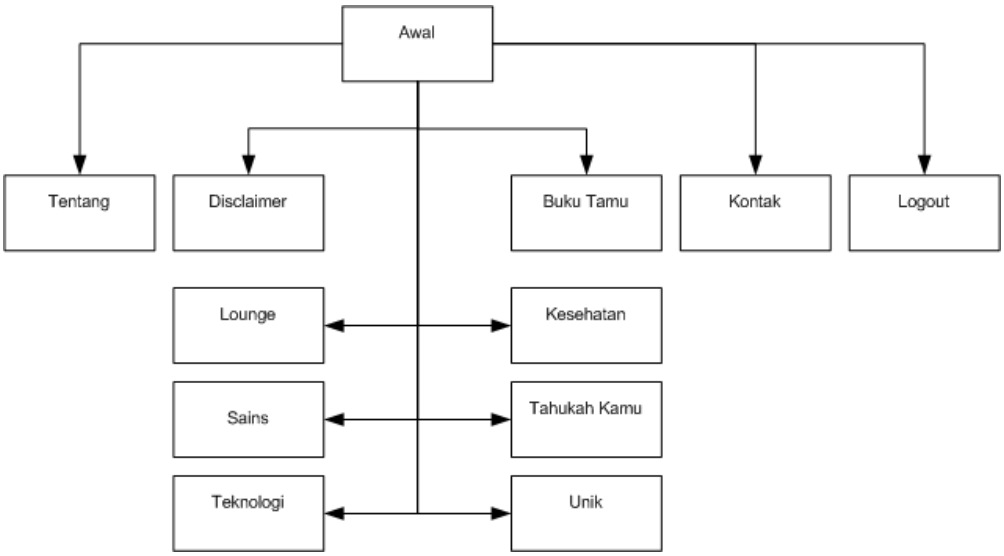
metode *SQL Injection* dan *Cross Site Scripting* sehingga menyebabkan aplikasi web terganggu.

3.2.2 Perancangan Antarmuka Target Simulasi

Berdasarkan hasil analisis yang telah diuraikan, maka berikut ini adalah perancangan antarmuka target simulasi.

3.2.2.1 Menu

Urutan menu-menu yang akan dirancang dalam bentuk Struktur Menu. Berikut adalah struktur menu yang dirancang.



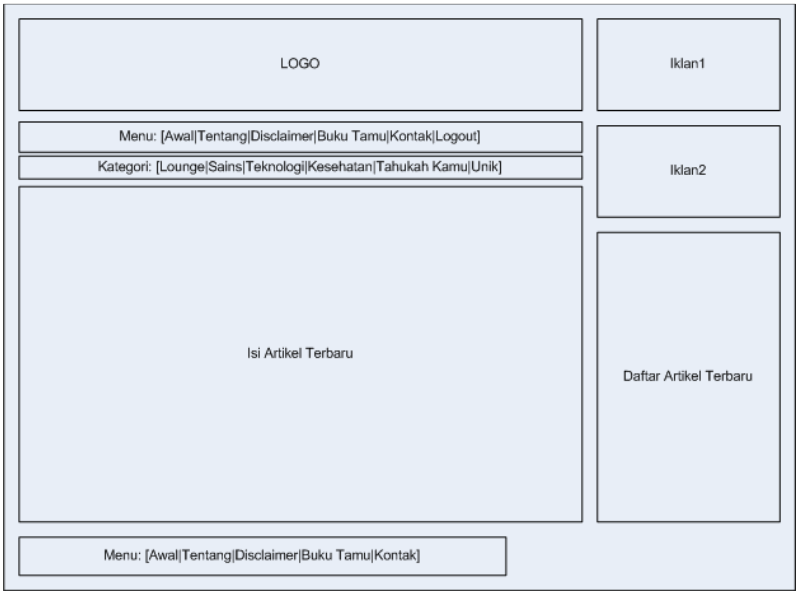
Gambar 3.39 *Struktur Menu Website*

3.2.2.2 Antarmuka

Tahap perancangan antarmuka dilakukan untuk merancang antarmuka dari sistem.

a. Halaman Indeks

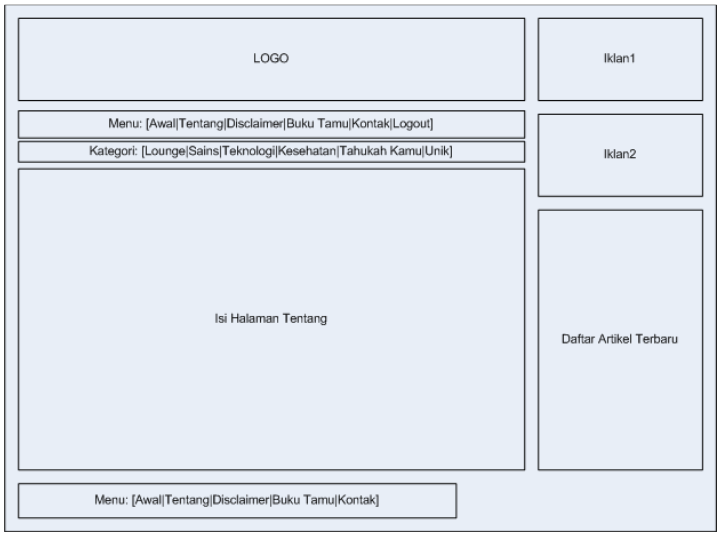
Halaman indeks adalah halaman pertama dari website informasimu.com. Halaman ini terdiri dari menu-menu, antara lain Awal, Tentang, Disclaimer, Buku Tamu, Kontak, Logout, Lounge, Sains, Teknologi, Kesehatan, Tahukah Kamu, Unik. Selain terdapat menu di atas, halaman indeks ini selalu menampilkan berita terbaru. Halaman indeks dapat dilihat pada Gambar 3.40.



Gambar 3.40 *Halaman Indeks*

b. Halaman Tentang

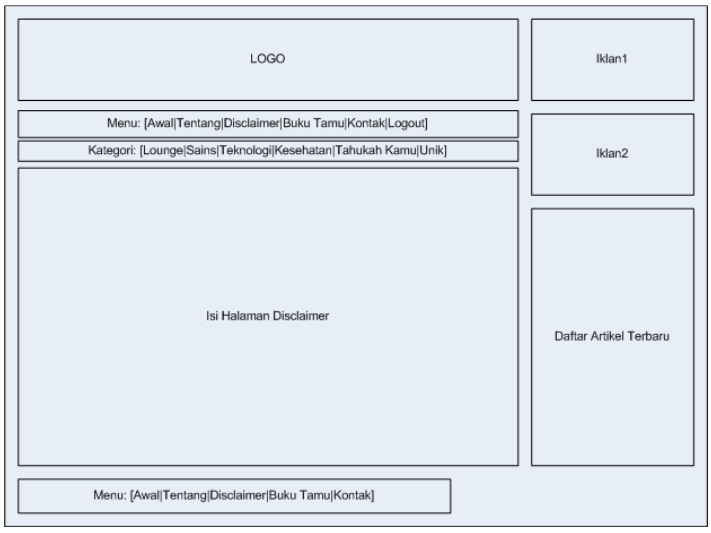
Halaman ini menjelaskan tentang website informasimu.com. Halaman Tentang dapat dilihat pada Gambar 3.41.



Gambar 3.41 *Halaman Tentang*

c. Halaman *Disclaimer*

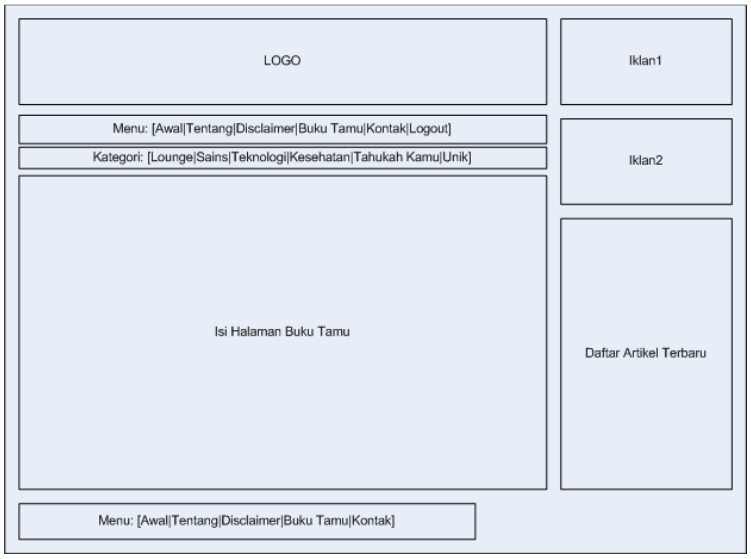
Halaman ini menjelaskan tentang penyalahgunaan artikel. Halaman Disclaimer dapat dilihat pada Gambar 3.42.



Gambar 3.42 *Halaman Disclaimer*

d. Halaman Buku Tamu

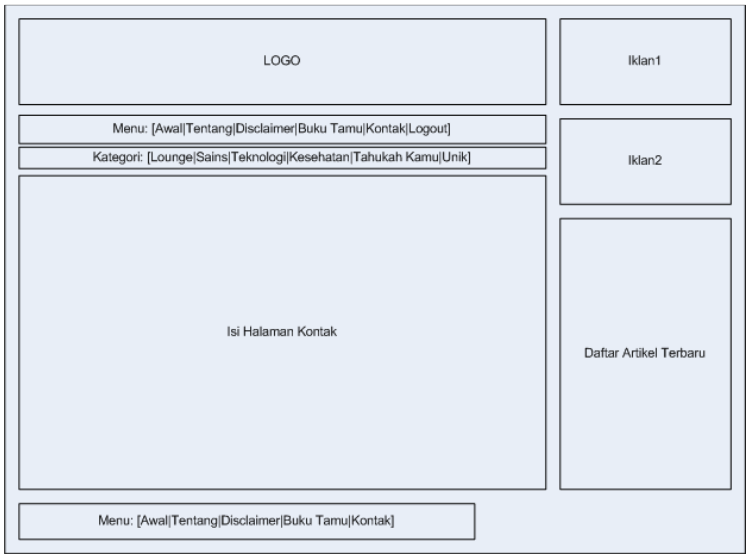
Halaman ini digunakan untuk mengirimkan pesan oleh pengunjung. Halaman Buku Tamu dapat dilihat pada Gambar 3.43.



Gambar 3.43 *Halaman Buku Tamu*

e. Halaman Kontak

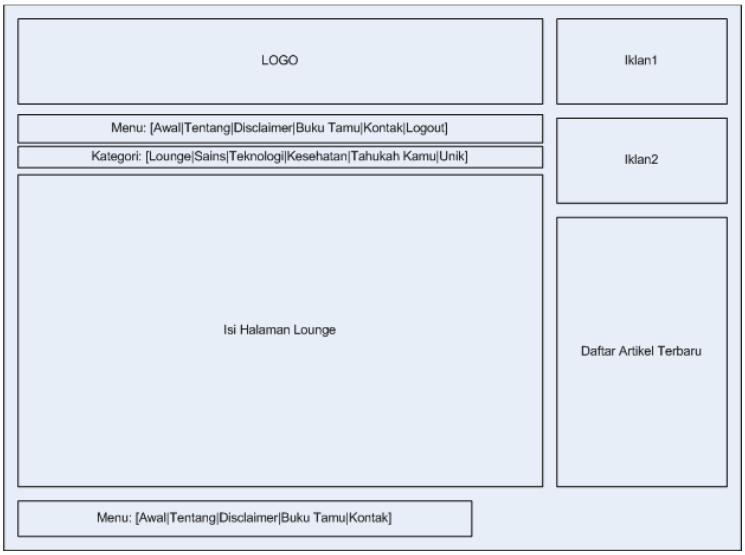
Halaman ini digunakan untuk menampilkan informasi kontak informasimu.com. Halaman Kontak dapat dilihat pada Gambar 3.44.



Gambar 3.44 *Halaman Kontak*

f. Halaman Lounge

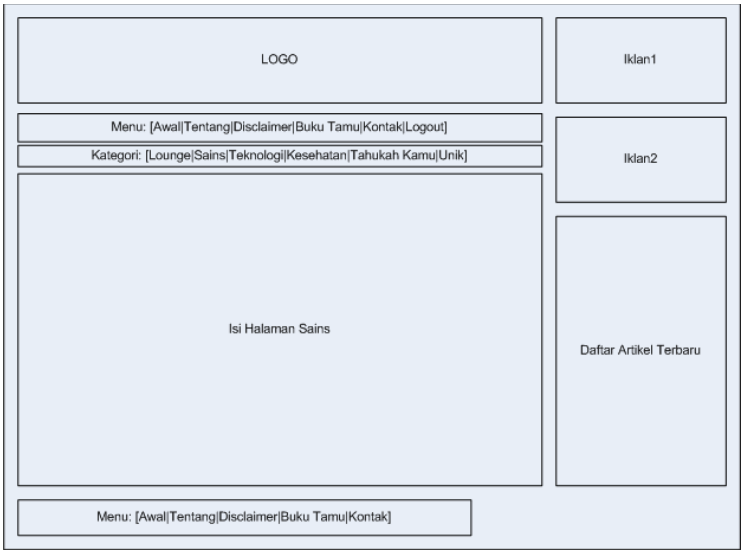
Halaman ini digunakan untuk menampilkan berita sesuai dengan kategori Lounge. Halaman Lounge dapat dilihat pada Gambar 3.45.



Gambar 3.45 *Halaman Lounge*

g. Halaman Sains

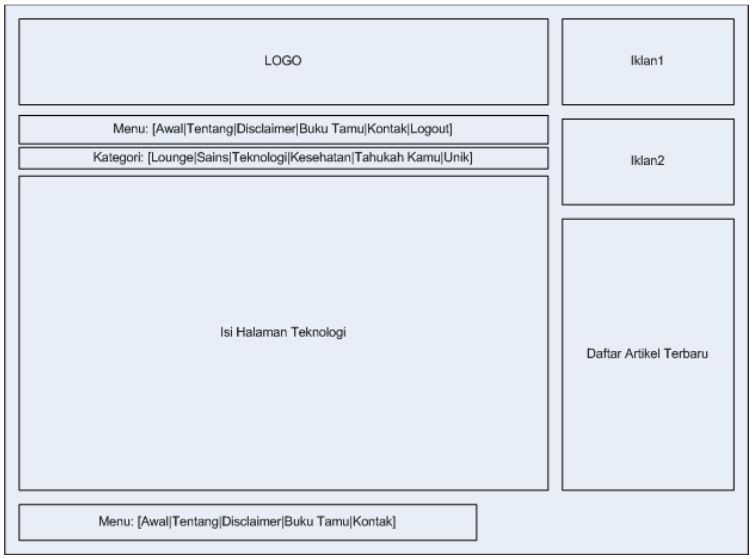
Halaman ini digunakan untuk menampilkan berita sesuai dengan kategori Sains. Halaman Sains dapat dilihat pada Gambar 3.46.



Gambar 3.46 *Halaman Sains*

h. Halaman Teknologi

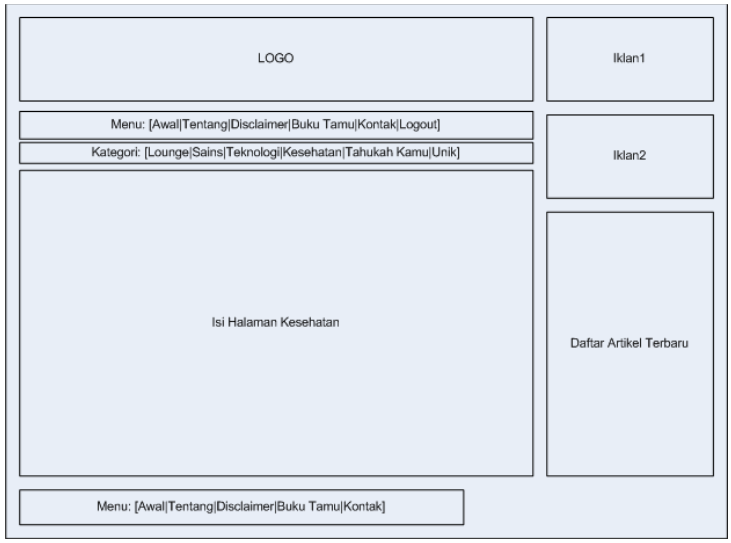
Halaman ini digunakan untuk menampilkan berita sesuai dengan kategori Teknologi. Halaman Teknologi dapat dilihat pada Gambar 3.47.



Gambar 3.47 *Halaman Teknologi*

i. Halaman Kesehatan

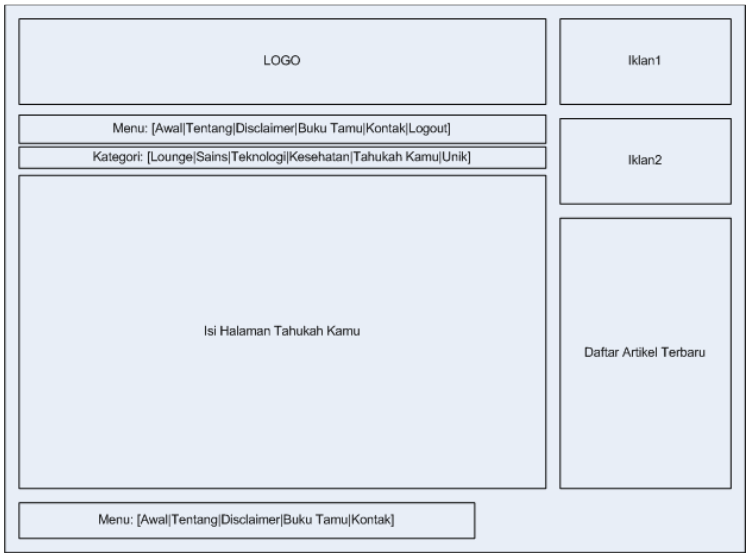
Halaman ini digunakan untuk menampilkan berita sesuai dengan kategori Kesehatan. Halaman Kesehatan dapat dilihat pada Gambar 3.48.



Gambar 3.48 *Halaman Kesehatan*

j. Halaman Tahukah Kamu

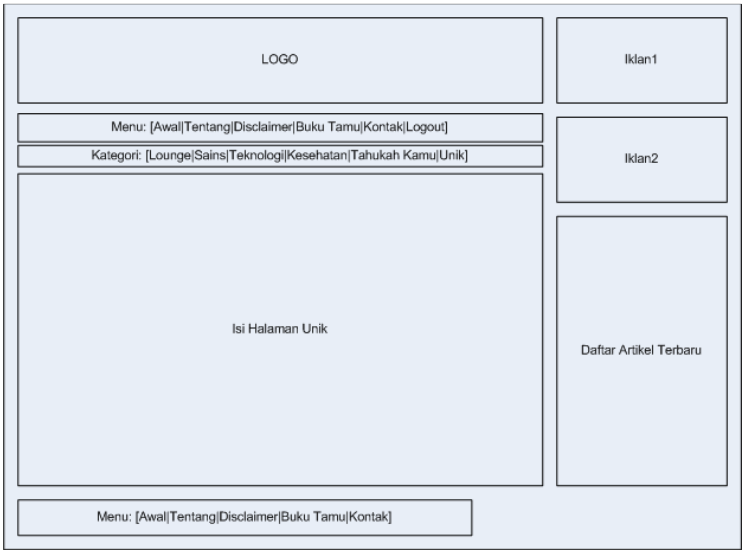
Halaman ini digunakan untuk menampilkan berita sesuai dengan kategori Tahukah Kamu. Halaman Tahukah Kamu dapat dilihat pada Gambar 3.49.



Gambar 3.49 *Halaman Tahukah Kamu*

k. Halaman Unik

Halaman ini digunakan untuk menampilkan berita sesuai dengan kategori Unik. Halaman Unik dapat dilihat pada Gambar 3.50.



Gambar 3.50 *Halaman Unik*