BAB II

LANDASAN TEORI

2.1 Website

Sejak internet diperkenalkan kepada dunia pada tahun 1972-1973, penggunaan internet semakin meluas, tidak hanya pada kalangan khusus (militer pada saat itu) saja. Seiring dengan perkembangannya, orang-orang yang memanfaatkan internet membuat sebuah sistem yang memudahkan pengaksesan internet oleh masyarakat luas. Sistem ini juga memungkinkan adanya peluang bisnis dalam bidang ini. Hal ini ditandai dengan didirikannya *provider* (penyedia layanan) internet sampai warnet (warung internet).

World Wide Web atau www atau singkatnya *web* memberikan gambaran secara jelas yang memudahkan untuk mencari dan melihat berbagai dokumen di internet, dokumen-dokumen di internet berhubungan baik satu sama lain, data atau halaman-halaman *web* agar terhubung pada komputer dimana saja dengan catatan komputer tersebut harus terhubung dengan *web* dan dapat mengakses internet. Secara mudah kita dapat mengidentifikasikan hubungan jarak jauh dengan mengklik *hyperlink* yang selalu berbeda warna dari teks biasa yang terdapat pada *web*.

Pada dasarnya website adalah sebuah cara untuk menampilkan sesuatu di internet, baik pribadi, perusahaan dan lain sebagainya. Website adalah sebuah tempat di internet, siapa saja di dunia ini dapat mengunjunginya, kapan saja dan dimana saja bisa berinternet maka situs tersebut dapat diakses.

Dengan adanya website maka pengunjung dapat melihat dan mengetahui tentang diri seseorang, barang, perusahaan atau apa saja yang ditampilkan di website tersebut.

Interaksipun terjadi di dalam website melalui pengiriman komentar seperti memberi pertanyaan, menjawab pertanyaan, memberikan kesan dan pesan ataupun membeli produk yang ditawarkan di website tersebut. Internet bagaikan sebuah pusat perdagangan terbesar di dunia dan website adalah salah satu toko atau kios di pusat perdagangan tersebut.

Mengapa seseorang atau perusahaan membutuhkan sebuah website?

- 5 tahun yang lalu memiliki sebuah website hanyalah untuk gengsi.
- 2 tahun yang lalu memiliki sebuah website hanyalah karena trendy.
- Hari ini memiliki sebuah website adalah keharusan.

Semua orang menggunakan website untuk segala sesuatu mulai dari pekerjaan pribadi, sekolah sampai dengan memesan makanan, karena prosesnya yang cepat, murah, mudah dan menyenangkan, bahkan yang suka belanja sekalipun mencari produk yang diinginkan di website.

2.2 Interaksi Dalam Sebuah Website

Dengan adanya website maka pengunjung dapat melihat dan mengetahui tentang diri seseorang, barang, perusahaan atau apa saja yang ditampilkan di website tersebut.

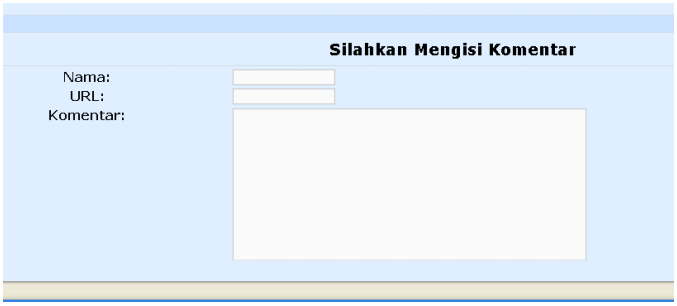
Interaksipun terjadi di dalam website melalui pengiriman komentar seperti memberi pertanyaan, menjawab pertanyaan, memberikan kesan dan pesan

ataupun membeli produk yang ditawarkan di website tersebut. Internet bagaikan sebuah pusat perdagangan terbesar di dunia dan website adalah salah satu toko atau kios di pusat perdagangan tersebut.

Website dapat digunakan untuk berbagai tugas, misalnya:

- Membuat pengumuman atau pemberitahuan.
- Memberikan pelayanan kepada Customer.
- Menerima masukan dari pengunjung atau pelanggan.
- Membagi dan mendistribusi data.
- Berkomunikasi langsung dengan pengunjung yang berada di belahan dunia.

Dengan adanya interaksi dalam website ini memang membuat website tersebut menarik dan mempermudah manusia dalam pekerjaannya, namun satu hal yang tidak kalah pentingnya adalah apakah website tersebut aman dari serangan bot-bot spam? Dengan adanya interaksi dalam website membuat website tersebut mempunyai celah untuk terserang spam dan bot-bot spam.



Gambar 2.1 Contoh pengisian komentar yang dapat dimanfaatkan oleh spammer

2.3 Spam

Spam pertama muncul pada tanggal 3 Mei 1978. Sampai saat ini spam dikenal sebagai anggota geng malware, email sampah, salah satu bahaya dari internet yang menduduki posisi teratas dalam catatan kejahatan komputer.

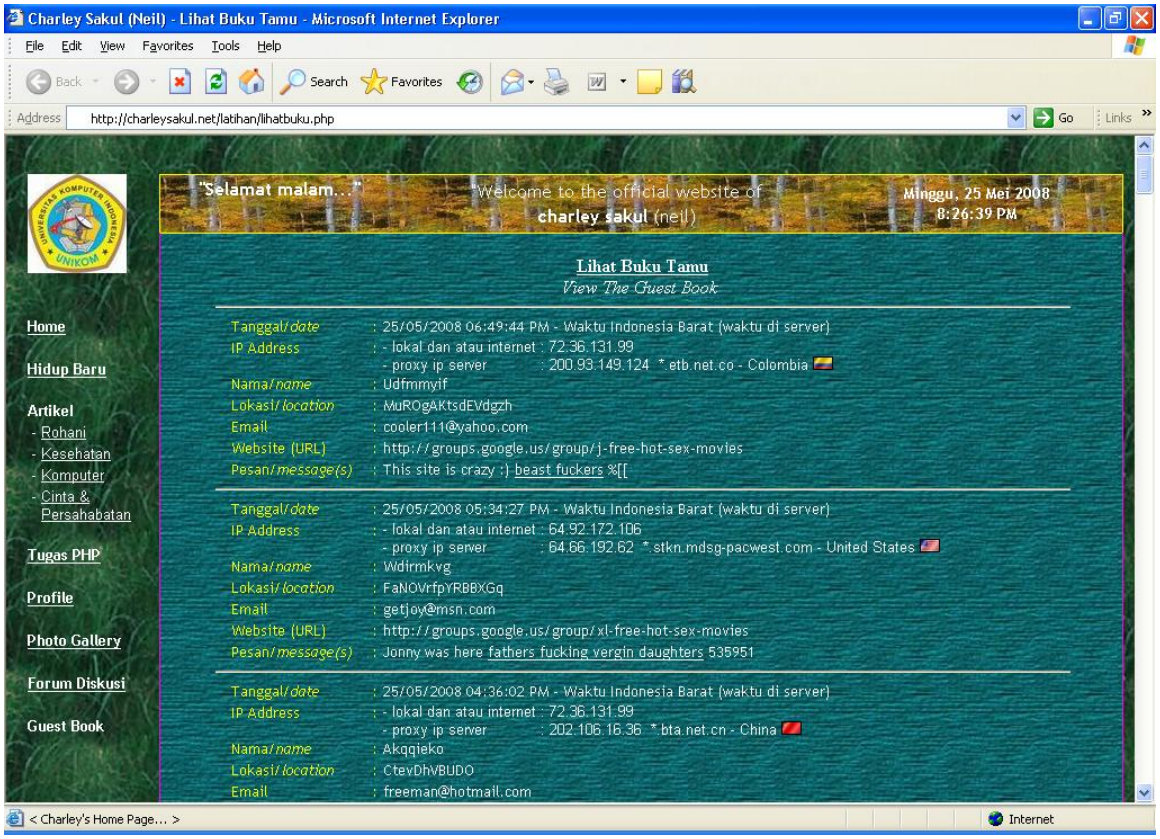
Sebuah blog Malaysia mengaku bahwa blognya dimasuki *Spammer*, sehingga dalam sehari ada 100 bot-bot spam yang masuk.

Spam adalah pengiriman informasi yang biasanya datang bertubi-tubi yang tidak pada tempatnya dan bersifat mengganggu untuk mempromosikan suatu barang atau jasa, bahkan untuk menyerang suatu website.

Spammer adalah orang yang mengirimkan spam atau melakukan *spamming*. Biasanya para *spammer* sudah membaca dan mengetahui website yang akan dijadikan target.

Bot adalah singkatan dari robot. Bot adalah program komputer yang dijalankan yang khusus dibuat untuk melakukan pekerjaan-pekerjaan otomatis. Kebanyakan bot dibuat oleh pembuat program yang sudah ahli dan sudah melewati serangkaian uji coba yang panjang.

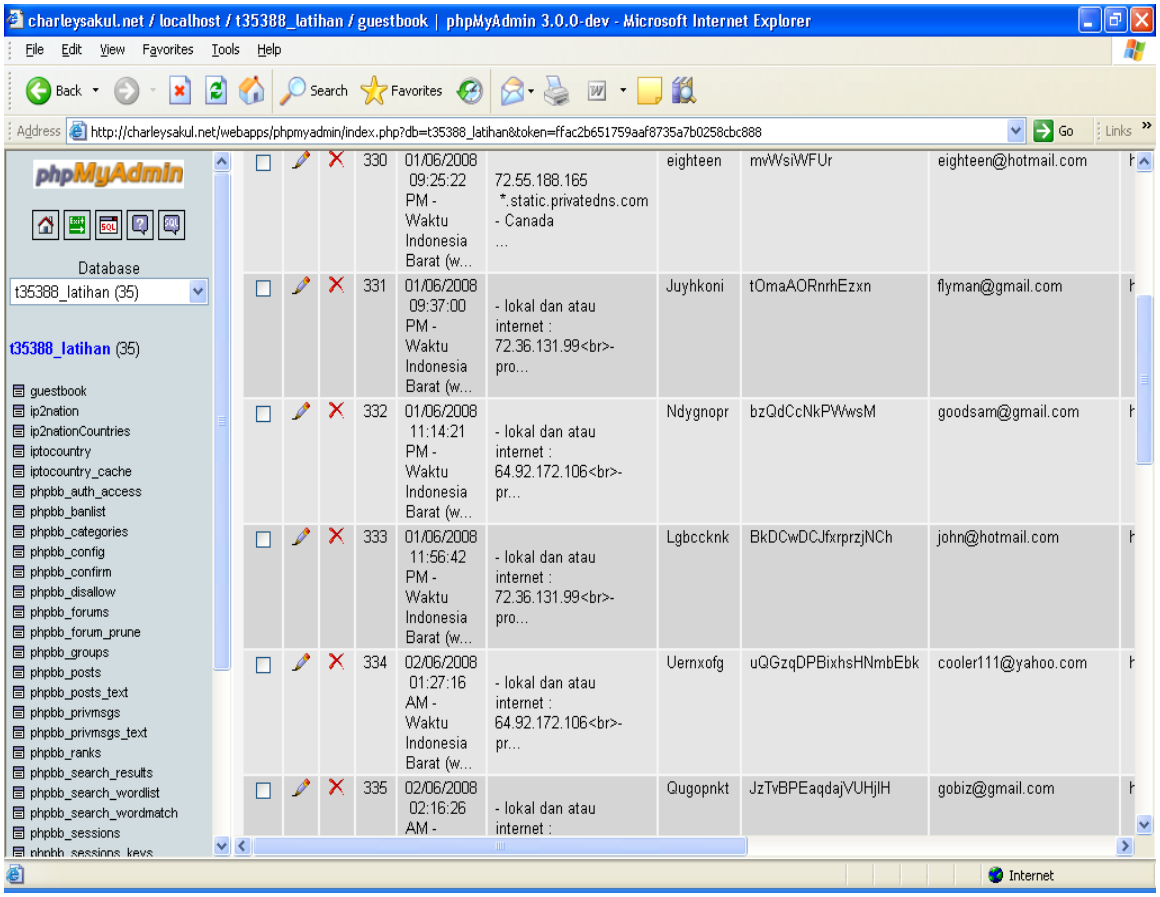
Bot spam adalah program komputer yang dibuat yang dijalankan untuk melakukan spam secara otomatis terhadap suatu website.



Gambar 2.2 Contoh website yang terserang spam dan bot-bot spam

Spam dan bot-bot spam yang telah berhasil masuk dalam suatu website kini tersimpan dalam database. Spam dan bot-bot spam ini sangat merugikan karena selain menimbulkan ketidaknyamanan bagi pengunjung dan pemilik website, juga memenuhi dan menghabiskan ruang (*space*) dalam database.

Bila belum ada penangkal untuk mengatasi permasalahan ini maka administrator website harus menghabiskan waktu untuk selalu online, menjaga dan menghapus semua spam dan bot-bot spam dari database.



Gambar 2.3 Contoh Bot-bot Spam Dalam Database

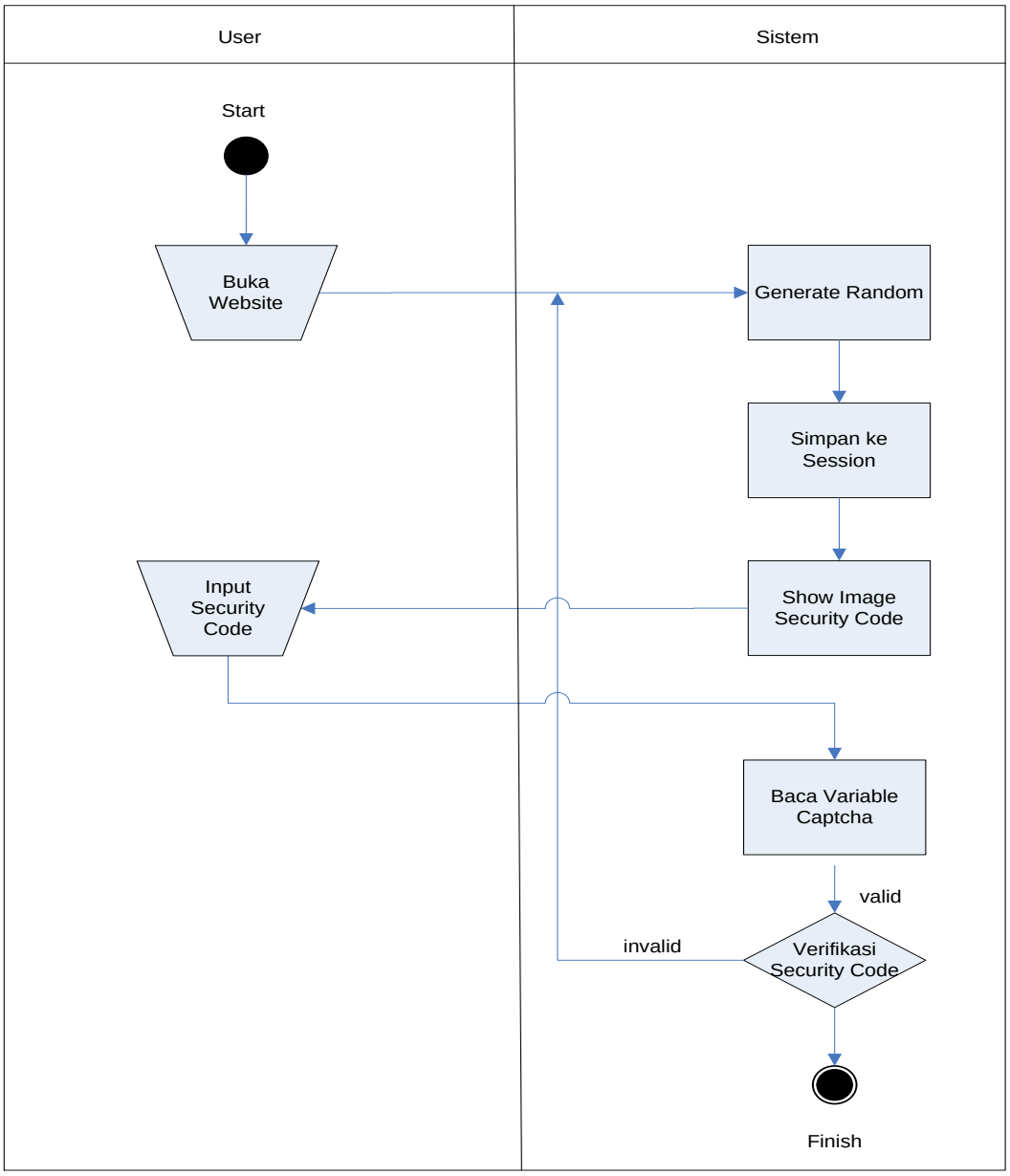
Sistem keamanan lama pada website menunjukkan bahwa website sering ditembusi spam baik pada saat registrasi maupun pada saat pengisian form untuk melakukan *posting*, seperti contoh pada pengiriman komentar dalam sebuah blog, forum diskusi PHPBB versi 2.0.x, pengisian buku tamu dan lain-lain. Untuk itu perlu dibuat suatu sistem keamanan website agar tidak ditembus oleh bot-bot spam.

2.4 Sistem Keamanan Website

Sistem keamanan website sangat diperlukan untuk menghalau atau memblokir spam dan bot-bot spam. Para *spammer* biasanya telah mengetahui kekurangan dari website yang akan dijadikan target atau sasaran. Bila sistem keamanan pada website masih minim atau bahkan tidak ada, maka ini adalah sasaran empuk para *spammer*, dimana *spammer* akan memasang bot-bot untuk menyerang website tersebut.

2.4.1 Generate Kode Keamanan

Captcha adalah *Completely Automated Public Turing test to tell Computers and Humans Apart"*. Captcha adalah sebuah tipe dari tes respon yang digunakan dalam perhitungan untuk meyakinkan bahwa jawaban (respon) tersebut tidak dihasilkan oleh sebuah computer. Captcha biasanya ditempatkan di form web untuk memastikan apakah pengguna itu manusia atau mesin. Tujuan dari captcha adalah memblok submission form dari spambot – skrip otomatis. Algoritma captcha yang paling bagus diciptakan oleh Carnegie Mellon University, dan algoritma tersebut digunakan oleh Yahoo, Google dan situs-situs besar lainnya.



Gambar 2.4 Flowchart Kode Keamanan Website

Konsep dari captcha ini banyak yang mengacu pada algoritma: generate random, simpan ke session, membaca variable captcha dari inputan user, mencocokan/validasi pada nilai session yang tersimpan, meloloskan sesuatu apabila sesuai, menggagalkan sesuatu apabila tidak sesuai.

Untuk membuat gambar dalam PHP tentu memanfaatkan GD Library dengan memanfaatkan session untuk mengirim nilai pada *security code* tersebut. Banyak

website yang menggunakan *security code* tetapi jawaban tersebut tersimpan di form, bisanya disimpan didalam <input type="hidden" />. Atau ada juga yang disimpan didalam link pada browsernya. Hal seperti itu sangat tidak aman, karena sama saja membuat *security code* dengan jawaban sekaligus. Sistem captcha yang baik adalah akan mengirim jawabannya dengan menggunakan session.

Berikut ini adalah contoh script untuk menghasilkan kode keamanan captha:

```
1. $result = md5(microtime() * 1000);
2. $result = substr($result, 0, 6);
3.
4. $im = imagecreate(90, 29);
5. $background_color = imagecolorallocate($im, 0xFF, 0x80, 0x00);
6. $text_color = imagecolorallocate($im, 0x02, 0x01, 0x00);
7. $line_color = imagecolorallocate($im, 0xFF, 0xFF, 0xFF);
8.
9. header("Content-type: image/png");
10. imagestring($im, 9, 5, 5, $result, $text_color);
11. imageline($im, 0, 0, 39, 29, $line_color);
12. imageline($im, 40, 4, 139, 29, $line_color);
13. imageline($im, 80, 4, 139, 98, $line_color);
14. imageline($im, 90, 4, 19, 8, $line_color);
15. imagepng($im);
```

Baris 1-2 adalah membuat karakter acak, microtime() * 1000 untuk membangkitkan bilangan acak, lalu masukkan bilangan acak tersebut ke fungsi md5() untuk mengacaknya lagi, lalu ambil 6 karakter dari depan dengan fungsi substr().

Baris 4-7 mempersiapkan imagenya beserta warna yang akan digunakan, imagecreate(90, 29) untuk membuat image, imagecolorallocate($im, 0xFF, 0x80, 0x00) untuk membuat warna yang akan digunakan, 0xFF, 0x80, 0x00 dan sebagainya itu adalah bilangan hexa dari sebuah warna.

Baris 9-15 ditampilkan imagenya, header("Content-type: image/png") untuk menampilkan image dengan keluaran png, imagestring($im, 9, 5, 5, $result, $text_color) untuk memasukkan karakter yang telah dibuat ke dalam image, imageline($im, 0, 0, 39, 29, $line_color) untuk membuat garis acak.

Captcha dapat dikembangkan sesuai kemauan dan kreasi dari pembuatnya, yaitu dengan membuat text acak dan acak juga letak kode kuncinya. Jadi ada 2 variabel untuk menyimpan text keseluruhan (tampil) dan text yg berisi kode. Untuk membedakan text kode dan text pengacak di tampilkan dengan warna yg berbeda, satu mencolok dan satu lebih lembut.

Tampilan seperti ini: **Qw3Rty** misalkan yg jadi text kode adalah warna biru (3Ry) dan text pengacak abu-abu (Qwt).

Jadi user harus memasukkan kode yg benar yaitu: **3Ry**

Berikut ini adalah scriptnya:

```
function generate($enc = ""){
// random char composition
$salt = "abcdefABCDEF1234567890";
$len = strlen($salt);
mt_srand(10000000*(double)microtime());

for ($i = 0; $i < $this->text_length; $i++){
if(($i == mt_rand($i,$this->text_length - 1)) && $this->key_length > 0){
$text = $salt[mt_rand(0,$len - 1)];
$this->text_result .= "" . $text . "";
$this->key_result .= $text;
$this->key_length--;
} else
$this->text_result .= "" . $salt[mt_rand(0,$len - 1)] . "";
}
// direct print code
print($this->text_result);
switch ($enc) {
case "md5":
return $this->catext_md5($this->key_result);
```

```
break;
case "sha1":
return $this->catext_sha1($this->key_result);
break;
default:
return trim($this->key_result);
 }
}
```

Ada juga yang mengembangkan captcha dengan model matematika. Contoh captcha matematika seperti di bawah ini.

Contoh tampilan adalah: **2+ 5**

User harus memasukkan hasil dari operasi matematika tersebut yaitu: **7**

Berikut ini adalah scriptnya:

```
/*
* Function to "translate" mathematic operation
* @val1: int
* @val2: int
* @op: int; if you add/remove case section, please update generate(); function
*/
function translate($val1, $val2, $op = 0){
switch ($op) {
case 0:
$this->_resultText = $val1 . ' + ' . $val2;
$this->_resultNumber = $val1 + $val2;
break;

case 1:
$this->_resultText = $val1 . ' - ' . $val2;
$this->_resultNumber = $val1 - $val2;
break;
default:
$this->translate($val1, $val2, 0);
 }
}

function generate($enc = ""){
$val1 = rand($this->_min, $this->_max);
$val2 = rand($this->_min, $this->_max);

// update this value if you modify(add/remove case) translate(); function
```

```
$opr = rand(0, 1);

$this->translate($val1, $val2, $opr);

print $this->_resultText;// printing code
switch ($enc) {
case "md5":
return $this->caMath_md5($this->_resultNumber);
break;
case "sha1":
return $this->caMath_sha1($this->_resultNumber);
break;
default:
return trim($this->_resultNumber);
  }
}
```

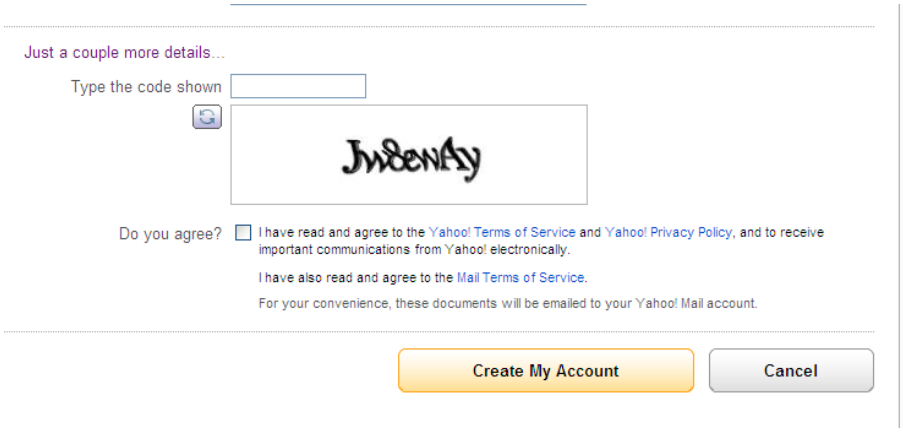
Captcha sering dibuat dengan tulisan yang miring, buram, bergaris, dan pada umumnya adalah gambar berisi text acak yang diberi distorsi sedemikian rupa. Ini mempunyai tujuan untuk memproteksi situs dari bot/robot agar bot spammer tidak bisa membacanya. Siapapun user yang memasukkan kode yang benar diasumsikan sebagai manusia. Jadi captcha digunakan untuk mencegah pengisian form secara otomatis oleh program tertentu yang disebut dengan bot atau robot.

OCR (Optical Character Recognation) adalah suatu program untuk membaca tulisan yang terdapat pada gambar. Captcha yang dibuat dengan tulisan yang miring, buram, bergaris, dan pada umumnya adalah gambar berisi text acak yang diberi distorsi sedemikian rupa sangat sulit dibaca oleh mesin program (robot), sehingga dapat diasumsikan bahwa yang mengisi form adalah manusia. Hal ini adalah untuk memproteksi form website dari spam.

2.4.2 Verifikasi Kode Keamanan

Verifikasi kode keamanan dibuat dalam bentuk *security image* atau *security code* atau *captcha* adalah suatu bentuk perlindungan terhadap suatu form dari serangan spam.

Proses verifikasi data kode keamanan adalah lebih diutamakan untuk memblokir bot, dimana diberikan kode keamanan yang harus diisi dengan benar. Contoh pada pengisian formulir pendaftaran untuk mendapat *account* email di yahoo, dimana pada bagian bawah diatas tombol *create my account* terdapat *security code* yang harus diisi.



Gambar 2.5 *Security code* pada form pendaftaran email yahoo

Security code yang tampil ini harus diisi dengan benar oleh semua user yang mendaftar, bila salah maka proses tidak akan dilanjutkan.

Cara kerja *security code* ini adalah:

1. Pertama:

   Form akan menampilkan *security code* yang di generate oleh script PHP, apabila halaman tersebut di refresh maka *security code* akan melakukan *generate* lagi sehingga kodenya berubah. Hal ini sangat berguna sekali untuk menangkal serangan spam.

2. Kedua:

   Setelah menekan tombol submit, script php akan memvalidasi data dan mencocokkan *security code* yang telah dimasukkan. Apabila *security code* tidak sama maka proses akan dibatalkan, apabila *security code* sama, maka proses akan dilanjutkan.

2.4.3 Verifikasi Komentar (Pemeriksaan dan Pendeteksian Spam)

Verifikasi komentar yang digunakan untuk pemeriksaan dan pendeteksian spam ini berisi filter dengan menggunakan algoritma Bayesian, dimana pengiriman komentar akan difilter apakah komentar tersebut spam atau bukan.

Bila data yang dimasukkan termasuk dalam kategori spam maka proses tidak akan dilanjutkan dan informasi bahwa data terdeteksi sebagai spam disampaikan kepada user. Bila data yang dimasukkan tidak termasuk dalam kategori spam maka proses data akan dilanjutkan untuk disimpan.

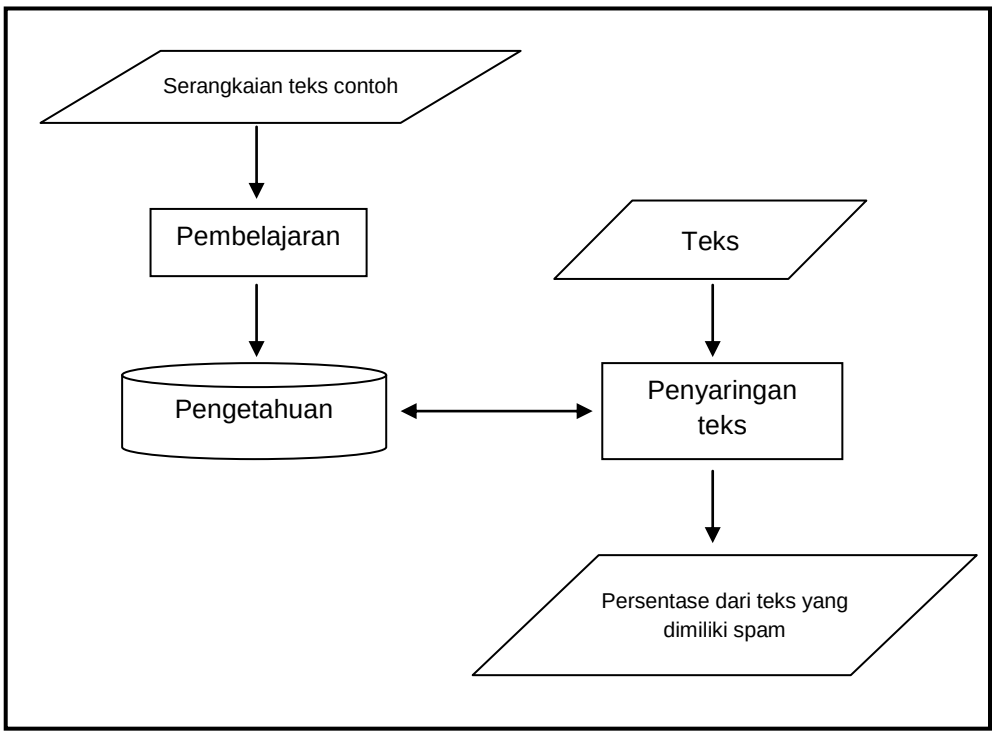
Pengecekan spam ini menggunakan tabel tersendiri yang menyimpan contoh-contoh komentar spam. Tabel tersebut harus terus diupdate oleh administrator, sehingga akan lebih menghasilkan suatu pemeriksaan spam yang lebih akurat lagi. Diperlukan juga sebuah tabel lain untuk menyimpan data yang bukan spam. Semua data yang tidak termasuk dalam kategori spam disimpan dalam tabel tersebut. Semua tabel ini dapat dilihat dan diubah oleh admin. User hanya bisa melihat report data, tidak bisa mengubah data.

2.5 Algoritma Bayesian

Dewasa ini telah dikembangkan berbagai algoritma untuk memfilter spam secara otomatis oleh perangkat lunak, baik di level MUA maupun MTA. Salah satu algoritma yang cukup handal untuk menghalau spam adalah algoritma Bayes. Algoritma ini bekerja memanfaatkan metode statistika Bayesian untuk mengklasifikasikan dokumen dalam katagori tertentu berdasarkan tingkat keterpercayaannya.

Algoritma Bayesian dinamai sesuai nama Thomas Bayes (1702-1761) yang pertama kali mengajukan persoalan statistik yang kini dikenal sebagai teorema Bayes. Istilah Bayesian sendiri baru muncul sekitar tahun 1950-an. Sebelumnya, teorema Bayes banyak dipakai dalam berbagai bidang, mulai dari perhitungan matematis orbit benda langit, statistika medis, hingga bidang hukum.

Teknik filter spam dengan metoda Bayesian (selanjutnya disebut filter Bayesian) pertama kali diajukan oleh M. Sahami, dkk pada 1998. Namun metoda ini baru mendapat perhatian pada 2002 ketika diulas oleh Paul Graham dalam sebuah paper. Selanjutnya, filter Bayesian mulai populer sebagai sebuah metode untuk membedakan spam dengan komentar regular. Dewasa ini filter Bayesian diimplementasikan pada berbagai perangkat lunak yang populer dan di perangkat filter server-side semacam SpamAssassin serta beberapa perangkat lunak mail server.



Gambar 2.6 Sistem Arsitektur Penyaringan Spam PHP Bayesian

1. Serangkaian teks contoh:

   adalah input komentar berupa serangkaian teks, dimana user memasukkan serangkaian atau sekumpulan teks berisi apa saja yang dapat dijadikan contoh pendeteksian spam.

2. Pembelajaran:

   Serangkaian teks contoh di atas kemudian dipelajari terlebih dahulu bagaimana isi teks-teks tersebut. Dalam mempelajari teks, perlu ditelusuri kata demi kata dalam rangkaian teks tersebut sehingga nanti dapat digolongkan mana spam dan mana yang bukan spam.

3. Pengetahuan:

   Setelah melalui tahap pembelajaran, kemudian masuk ke tahap pengetahuan, dimana mulai bisa diprediksi apakah teks-teks tersebut tergolong sebagai spam atau bukan. Pengetahuan yang didapat untuk memprediksi spam dan bukan spam ini kemudian dilanjutkan ke penyaringan teks.

4. Penyaringan teks:

   Serangkaian teks tersebut kemudian melalui tahap penyaringan. Dalam tahap penyaringan ini, teks akan diperiksa dan dideteksi seberapa besar persentase spam dalam rangkaian teks tersebut.

   Pemeriksaan dan pendeteksian spam dibantu dengan text yang benar-benar spam. Kemudian pada penyaringan teks ini akan dibandingkan sekumpulan teks tadi dengan teks yang benar-benar spam.

   Dengan demikian akan didapat berapa persentase sekumpulan atau serangkaian teks tadi yang mengandung spam.

5. Teks:

   adalah teks yang benar-benar spam yang akan digunakan sebagai perbandingan untuk membandingkan sekumpulan teks yang akan dideteksi sehingga akan diperoleh berapa persentase spam dari sekumpulan teks yang dideteksi tersebut.

6. Persentase teks yang dimiliki spam:

   adalah hasil yang didapat setelah melalui tahap penyaringan, yaitu berapa persentase serangkaian teks yang mengandung spam setelah melalui pemeriksaan dan pendeteksian spam.

Bayesian filter merupakan metode terbaru yang digunakan untuk mendeteksi *spam*. Algoritma ini memanfaatkan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi probabilitas di masa depan berdasarkan pengalaman di masa sebelumnya. Dua kelompok peneliti, satu oleh Pantel dan Lin, dan yang lain oleh Microsoft Research memperkenalkan metode statistik Bayesian ini pada teknologi anti spam filter. Tetapi yang membuat algoritma Bayesian filtering ini popular adalah pendekatan yang dilakukan oleh Paul Graham.

2.5.1 Tabel dan Field Pada Bayesian Filter

Pada Bayesian filter membutuhkan suatu pembelajaran sehingga dibuatlah tabel untuk pembelajaran dengan nama tabel knowledge_base yang berisikan 4 field yaitu ngram, belongs, repite dan percent.

- Ngram

Ngram adalah sebuah tipe dari suatu model probabilitas atau penentuan kemungkinan untuk memprediksi sesuatu item berikutnya (dalam hal ini adalah komentar) apakah termasuk dalam kategori spam atau bukan spam.

Field ngram ini digunakan Bayesian filter untuk menentukan probabilitas atau kemungkinan suatu komentar, yaitu berapa besar komentar yang dideteksi atau dicek termasuk dalam kategori spam atau bukan spam.

- Belongs

Belongs adalah sebuah field yang menampung informasi mengenai suku kata, yang mana suku kata tersebut termasuk spam atau bukan spam.

- Repite

Repite berasal dari bahasa Spanyol *(spanish)* yang bila diterjemahkan ke bahasa Inggris adalah *"repeat"* yang artinya dalam bahasa Indonesia adalah mengulangi. Pada bayesian filter ini, field ini digunakan untuk menampung informasi mengenai data jumlah kata/suku kata yang sudah muncul/berulang beberapa kali.

- Percent

Percent adalah sebuah field yang akan menunjukkan dalam persen hasil dari pendeteksian Bayesian filter ini.

Bayesian filter membutuhkan juga sebuah tabel lagi untuk digunakan sebagai sampel atau contoh, yaitu untuk membandingkan suatu komentar yang masuk dengan komentar yang sudah dikategorikan sebagai spam, untuk itu dibuat sebuah tabel dengan nama tabel examples. Tabel examples berisikan 2 field yaitu text dan state.

- Text

Field ini berisikan atau digunakan untuk menampung semua data-data sampel atau contoh komentar spam. Data-data sampel atau contoh komentar yang bukan spam

juga ditampung disini. Untuk memisahkan data spam dan data bukan spam dibutuhkan sebuah field yang dinamakan field state.

- State

Field ini digunakan untuk menentukan atau memisahkan data atau komentar yang dikategorikan sebagai spam dan bukan spam.

2.5.2 Class Pada Bayesian Filter

Seperti yang telah dijelaskan di atas pada gambar 2.6 sistem arsitektur penyaringan spam php Bayesian, Bayesian filter ini menggunakan 3 class yaitu: trainer, ngram dan spam.

- Trainer

Pada class trainer ini digunakan Bayesian filter untuk pembelajaran yaitu suatu komentar dipelajari dulu oleh sistem dengan cara dibandingkan dengan komentar spam yang telah ada yang disimpan dalam tabel examples, sehingga suatu komentar yang dicek atau dideteksi dapat dikategorikan termasuk dalam kategori spam atau bukan spam.

- Ngram

Pada class ngram ini digunakan Bayesian filter untuk menentukan probabilitas atau kemungkinan suatu komentar, yaitu berapa besar komentar yang dideteksi atau dicek termasuk dalam kategori spam atau bukan spam.

- Spam

Class ini adalah class spam dimana komentar tersebut termasuk atau terdeteksi sebagai spam.

2.5.3 Pendeteksian Spam Pada Bayesian Filter

Bayesian filter mendeteksi spam dengan cara menghitung probabilitas dari suatu pesan (komentar) berdasarkan isinya. Probabilitas ini dapat dihitung melalui suatu database. Kemudian dengan suatu metode training, software anti spam yang menggunakan algoritma Bayesian dapat dilatih untuk melihat kata-kata yang sering digunakan pada komentar spam, sehingga pada akhirnya dihasilkan filter anti spam yang akurat dengan sesedikit mungkin *false positives*. *False positives* adalah komentar legal yang ditujukan kepada penerima, tetapi karena kesalahan dari filter anti spam, dikategorikan menjadi komentar spam.

Program MOD yang dibuat adalah untuk filterisasi. Filterisasi ini sangat berhubungan dengan algoritma Bayesian. Algoritma Bayesian adalah untuk mendapatkan data probabilitas untuk dimasukkan nanti ke bagian filterisasi.

Pada awalnya, Bayesian filter ini harus dicoba terlebih dahulu menggunakan sejumlah komentar spam dan bukan spam. Bayesian filter akan menghitung probabilitas lokal dari suatu kata, misalnya kata "*adult*", untuk muncul di kelompok komentar spam. Probabilitas lokal ini dapat dirumuskan seperti berikut:

$$P_{local-spam} = N_{spam} / (N_{spam} + N_{non\text{-}spam})$$

dimana : $P_{local-spam}$ = probabilitas suatu kata "x" terdapat pada komentar spam
N_{spam} = jumlah komentar spam dengan kata "x" di dalamnya
$N_{non\text{-}spam}$ = jumlah komentar non-spam dengan kata "x" di dalamnya

Contohnya adalah kata "free" terdapat dalam 70 spam dan 10 bukan spam, maka kemungkinan (probabilitas) kata "free" adalah 70 / (70 + 10) = 70/80 = 0,875

berarti kata “free” mempunyai probabilitas yang tinggi untuk masuk ke dalam kategori spam.

2.6 Definisi Pemrograman Berorientasi Objek

Pemrograman berorientasi objek, atau sering juga disebut dengan OOP (*Object Oriented Programming*), merupakan paradigma baru dalam pengembangan sistem/perangkat lunak. Tidak seperti pendahulunya, yaitu pemrograman dengan teknik terstruktur (*structured programming*) yang sering mengalami kegagalan dalam hal sistem/perangkat lunak yang tidak sesuai dengan kebutuhan dan harapan pengguna (*user’s need and expectations*) serta kerap kali tidak dapat memenuhi jadwal yang telah ditentukan sebelumnya. Pemrograman berorientasi objek dengan paradigma barunya, menjanjikan tingkat penggunaan ulang (*reusability*), kinerja (*performance*), serta kehandalan (*reliability*) yang lebih tinggi. (Roger Pressman, 2000).

Saat ini, dunia pemrograman berorientasi objek mengenal UML (*Unified Modelling Language*) sebagai alat utama dalam analisis dan perancangan sistem. UML merupakan alat yang sangat sesuai dengan pemrograman berorientasi objek sebab konsep dasarnya adalah memodelkan kelas-kelas (beserta atribut serta operasi di dalamnya) bersamaan dengan relasi-relasi yang terjadi antar kelas yang bersangkutan.

2.7 Definisi UML (*Unified Modeling Language*)

UML (*Unified Modelling Language*) adalah sebuah "bahasa" yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemaduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan oleh XML. Standar UML dikelola oleh OMG (*Object Management Group*).

UML (*Unified Modelling Language*) adalah bahasa pemodelan untuk menspesifikasikan, memvisualisasikan, membangun dan mendokumentasikan artifak-artifak dari sistem.

1. Di dalam *system intensive process*, metode diterapkan sebagai proses untuk menurunkan atau mengevolusikan sistem.
2. Sebagai bahasa, UML digunakan untuk komunikasi yaitu alat untuk menangkap pengetahuan mengenai satu subyek dan mengekspresikan pengetahuan (sintaks) yang memperdulikan subyek untuk maksud komunikasi. Subyek adalah sistem yang dibahas.

3. Sebagai bahasa pemodelan, UML fokus pada pemahaman subyek melalui formulasi model dari subyek. Model memuat pengetahuan pada subyek, dan aplikasi dari pengetahuan ini berkaitan dengan intelejensia.
4. Berkaitan dengan unifikasi.UML memadukan praktek rekayasa terbaik sistem informasi dan industri, meliputi beragam tipe sistem (perangkat lunak dan non perangkat lunak), domain (bisnis, perangkat lunak) dan proses siklus hidup.
5. begitu diterapkan untuk menspesifikasikan sistem, UML dapat digunakan untuk mengkomunikasikan "apa" yang diperlukan dari sistem dan "bagaimana" sistem dapat direalisasikan.
6. Begitu diterapkan untuk menvisualisasikan sistem, UML dapat digunakan untuk menjelaskan sistem secara visual sebelum direalisasikan.
7. Begitu diterapkan untuk membangun sistem, UML dapat digunakan untuk memandu realisasi sistem serupa dengan "blueprint".
8. Begitu diterapkan untuk mendokumentasikan sistem, UML dapat digunakan untuk menangkap pengetahuan mengenai sistem pada seluruh siklus hidup.

Dengan melihat definisi UML diatas, maka UML bukanlah bahasa pemrograman visual, tapi bahasa pemodelan visual. UML bukanlah spesifikasi kakas, tapi spesifikasi bahasa pemodelan. UML juga bukan proses, tapi yang memungkinkan proses-proses.

Tujuan utama perancangan UML adalah :

1. Menyediakan bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model-model yang berarti.
2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep-konsep inti.
3. Mendukung spesifikasi independen bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman bahasa pemodelan.
5. Mendorong pertumbuhan pasar kakas berorientasi objek.
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, *framework* dan *pattern.*

2.7.1 Diagram Teknik Pemodelan UML

Diagram mengemukakan banyak hal, penggunaan notasi yang terdefinisi baik dan ekspresif adalah penting pada proses pengembangan perangkat lunak, yaitu :

1. Notasi standar memungkinkan pengembang mendeskripsikan skenario atau rumusan arsitektur dan kemudian mengkomunikasikan secara tidak ambigu.
2. Notasi yang bagus membebaskan otak untuk berkonsentrasi pada masalah-masalah yang lebih lanjut.
3. Notasi yang baik memungkinkan mengeleminasi keperluan pemeriksaan konsistensi dan kebenaran keputusan keputusan dengan menggunakan *tool* terotomatisasi.

2.7.2 Diagram Struktur

Diagram ini untuk memvisualisasi, menspesifikasikan, membangun dan mendokumentasikan aspek statik dari sistem. Diagram struktur di UML terdiri dari:

1. Diagram kelas (*Class diagram*)

   Diagram ini menunjukkan sekumpulan kelas, *interface* dan kolaborasi dan keterhubungannya.

2. Diagram objek (*Object diagram*)

   Diagram ini menunjukkan sekumpulan objek dan keterhubungannya.

3. Diagram komponen (*Component diagram*)

   Diagram ini menunjukkan organisasi dan kebergantungan di antara sekumpulan komponen. Diagram ini merupakan pandangan statik terhadap implementasi sistem.

4. Diagram deployment (*Deployment diagram*)

   Diagram ini menunjukkan konfigurasi pemrosesan saat jalan dan komponen komponen yang terdapat didalamnya.

2.7.3 Diagram Perilaku

Diagram ini untuk memvisualisasi, menspesifikasi, membangun dan mendokumentasikan aspek dinamis dari sistem. Diagram perilaku di UML terdiri dari :

1. Diagram *use-case* (*Use case diagram*)

   Diagram ini menunjukkan sekumpulan kasus fungsional dan *actor* (jenis kelas khusus) dan keterhubungannya.

2. Diagram sekuen (*Sequence diagram*)

   Diagram ini menunjukkan interaksi yang terjadi antar objek. Diagram ini merupakan pandangan dinamis terhadap sistem.

3. Diagram kolaborasi (*Collaboration diagram*)

   Diagram ini juga merupakan diagram interaksi. Diagram ini menekankan pada organisasi struktur dari objek-objek yang mengirim dan menerima pesan.

4. Diagram *statechart* (*Statechart diagram*)

   Diagram ini adalah *state-machine* diagram, berisi state, transisi, kejadian dan aktivitas. Statechart merupakan pandangan dinamis dari sistem.

5. Diagram aktivitas (*Activity Diagram*)

   Diagram ini untuk menunjukkan aliran aktivitas di sistem. Diagram ini adalah pandangan dinamis terhadap sistem.

2.7.4 Notasi dalam UML (*Unified Modeling Language*)

UML menyediakan beberapa notasi dan artifak standar yang bisa digunakan sebagai alat komunikasi bagi para pelaku dalam proses analisis dan desain. Artifak di dalam UML didefinisikan sebagai informasi dalam berbagai bentuk yang digunakan atau dihasilkan dalam proses pengembangan perangkat lunak.

1. *Actor*

Actor adalah segala sesuatu yang berinteraksi dengan sistem aplikasi komputer. Jadi *actor* ini bisa berupa orang, perangkat keras, atau mungkin objek lain dalam sistem yang sama. Biasanya yang dilakukan *actor* adalah memberikan informasi pada sistem dan atau memerintahkan sistem untuk melakukan sesuatu.

2. *Class*

Class merupakan pembentuk utama dari sistem berorientasi objek karena *class* menunjukkan kumpulan objek yang memiliki atribut dan operasi

yang sama. *Class* digunakan untuk mengabstraksikan elemen-elemen dari sistem yang sedang dibangun.

3. *Interface*

Interface merupakan kumpulan operasi tanpa implementasi dari suatu *class*.

4. *Use Case*

Use case menjelaskan urutan kegiatan yang dilakukan *actor* dan sistem untuk mencapai suatu tujuan tertentu walaupun menjelaskan kegiatan. Namun *use case* hanya menjelaskan "apa" yang dilakukan oleh *actor* dan sistem, bukan "bagaimana" *actor* dan sistem melakukan kegiatan tersebut.

5. *Interaction*

Interaction digunakan untuk menunjukkan baik aliran pesan atau informasi antar obyek maupun hubungan antar objek.

6. *Package*

Package adalah kontainer atau wadah konseptual yang digunakan untuk mengelompokkan elemen-elemen dari sistem yang sedang di bangun, sehingga bisa dibuat model yang lebih sederhana. Tujuannya adalah untuk mempermudah penglihatan (*visibility*) dari model yang sedang dibangun.

7. *Note*

Note digunakan untuk memberikan keterangan dan komentar tambahan dari satu elemen sehingga bisa langsung terlampir dalam model. *Note* ini bisa ditampilkan ke semua elemen notasi yang lain.

8. *Dependency*

Dependency merupakan relasi yang menunjukkan bahwa perubahan pada salah satu elemen memberi pengaruh pada elemen lain.

9. *Boundary Class*

Boundary class adalah *class* yang menghubungkan *user* dengan sistem.

10. *Control Class*

Control class adalah *class* yang mengkoordinasi aktivitas dalam sistem. *Class* ini menghubungkan *boundary class* dengan *entity class*.

11. *Entity Class*

Entity Class adalah *class* yang menghubungkan dengan data atau informasi yang digunakan oleh sistem. *Entity class* ini adalah *class* yang menyimpan dan mengelola data dapat dikombinasikan.

2.8 Software Pendukung

Diperlukan beberapa software pendukung agar sistem keamanan website dapat dibuat dengan baik.

2.8.1 PHP

PHP atau (*Personal Home Page*) adalah bahasa *server - side scripting* yang menyatu dengan HTML untuk membuat halaman web yang dinamis. Maksud dari *server - side scripting* adalah sintaks dan perintah-perintah yang diberikan akan sepenuhnya dijalankan di server, tetapi disertakan pada dokumen HTML.

Pembuatan *web* ini merupakan kombinasi antara PHP sendiri sebagai bahasa pemprograman dan HTML sebagai pembangun halaman *web*. Ketika seorang pengguna internet akan membuka suatu situs yang menggunakan fasilitas *server - side scripting* PHP, maka terlebih dahulu server yang bersangkutan akan memproses semua perintah PHP di server lalu mengirimkan hasilnya dalam format HTML ke *web browser* pengguna internet tadi. Dengan demikian seorang pengguna internet tidak dapat melihat kode program yang ditulis dalam PHP sehingga keamanan dari halaman *web* menjadi lebih terjamin.

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa script sejenis. PHP difokuskan pada pembuatan script server-side, yang bisa melakukan apa saja yang dapat dilakukan oleh CGI, seperti mengumpulkan data dari form, menghasilkan isi halaman web dinamis, kemamapuan mengirim serta menerima cookies, bahkan lebih daripada kemampuan CGI.

PHP dapat digunakan pada semua sistem operasi, antara lain Linux, Unix (termasuk variannya HP-UX, Solaris, dan OpenBSD), Microsoft Windows, Mac OS X, RSIC OS. PHP juga mendukung banyak Web Server, seperti Apache, Microsoft Internet Information Server (MIIS), Personal Web Server (PWS), Netscape and iPlanet servers, Oreilly Website Pro Server, Audium, Xitami, OmniHTTPd, dan masih banyak lagi lainnya, bahkan PHP dapat bekerja sebagai suatu CGI processor.

PHP tidak terbatas pada hasil keluaran HTML (HyperText Markup Language). PHP juga memiliki kemampuan untuk mengolah keluaran gambar, file

PDF, dan movies Flash. PHP juga dapat menghasilkan teks seperti XHTML dan file XML lainnya.

Salah satu fitur yang dapat diandalkan oleh PHP adalah dukungan terhadap database, seperti Adabas D, dBase, Direct MS-SQL, Empress, FilePro (read only), FrontBase, Hyperwave, IBM DB2, Informix, Ingres, Interbase, MSQL, MySQL, ODBC, Oracle (OC17 dan OC18), Ovrimos, PostgrSQL, Solid, Sybase, Unix DBM, Volacis.

Sintaks program/script PHP ditulis dalam apitan tanda khusus PHP. Asda empat macam pasangan tag PHP yang dapat digunakan untuk menandai blok script PHP:

<?php

Isi Script

?>

atau

<script language = “PHP”>

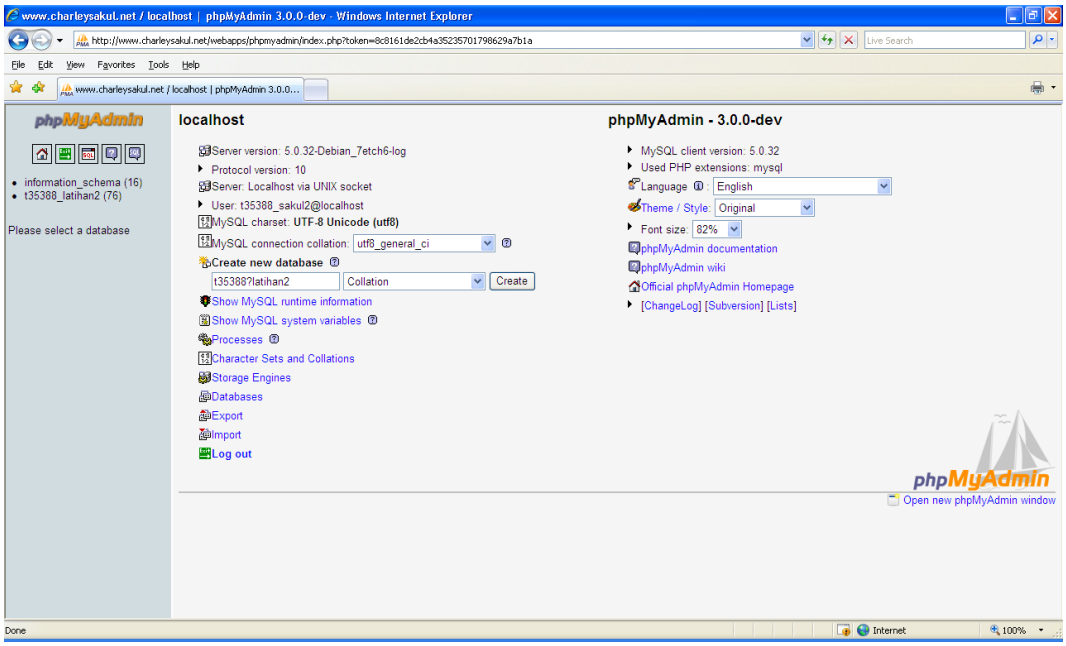
Isi Script

</script>

2.8.2 Database MySQL

MySQL adalah multi *user database* yang menggunakan bahasa *Structured Query Language* (SQL). MySQL dalam operasi *client* - *server* melibatkan *server* daemon MySQL di sisi server dan berbagai macam program serta *library* yang berjalan di sisi *client*. MySQL mampu menangani data yang cukup besar.

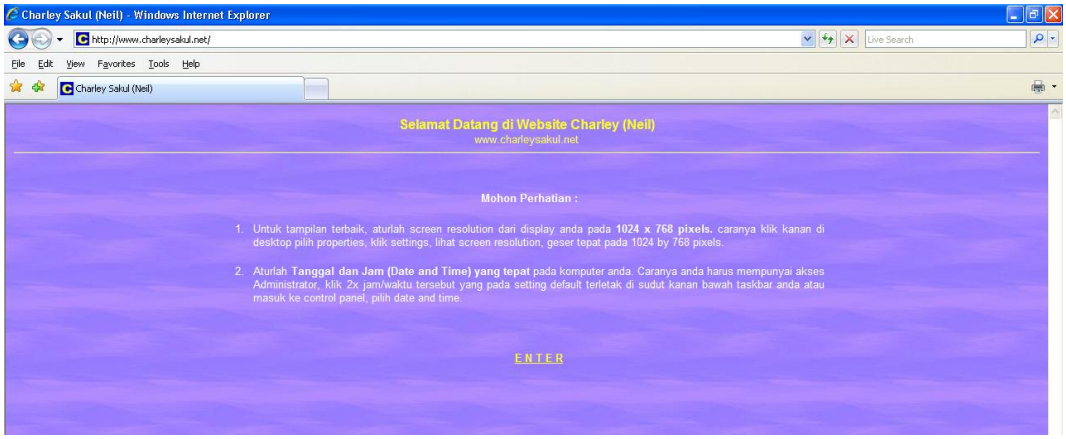
Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 *database*, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 Gigabyte data. SQL adalah bahasa standar yang digunakan untuk mengakses *database server*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai standar industri. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user - friendly* dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah - perintah pemrograman. MySQL merupakan *software database* yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performa *query* dari *database* nya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. MySQL ini juga sudah dapat berjalan pada lingkungan Windows.



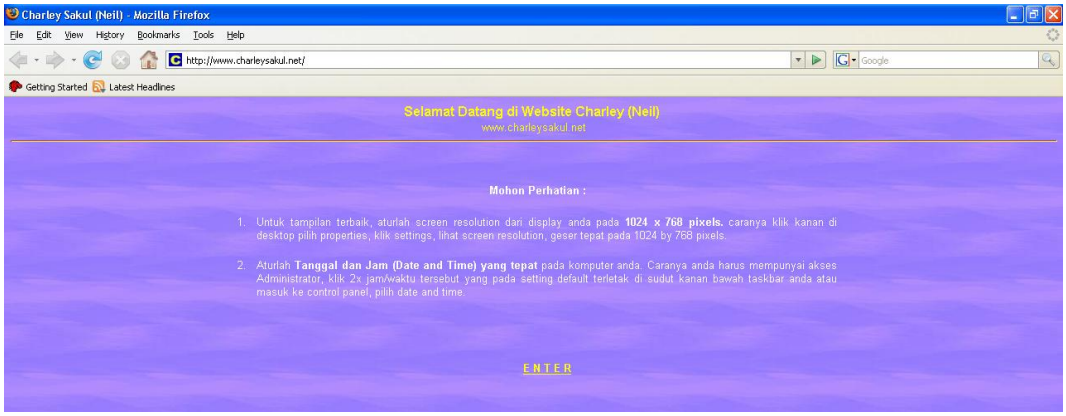
Gambar 2.7 Contoh Database MySQL

2.8.3 Web Browser

Web Browser adalah sebuah aplikasi yang digunakan untuk menampilkan file-file yang mendukung web yang berekstensi *.htm, *.html, *.php dan masih banyak lainya, dan juga beberapa *file image* (gambar). Melalui *web browser* penggna internet dapat memanfaatkan fasilitas internet yang ada pada sebuah *server* internet dan berinteraksi dengan pengguna internet lainya di sleuruh dunia, beberapa *browser* yang cukup dikenal diantaranya yaitu Internet Explorer, Mozilla firefox, opera, Netscape Navigator dll.



Gambar 2.8 Contoh web browser Internet Explorer



Gambar 2.9 Contoh web browser Firefox

2.8.4 *Web Server Apache*

Web server atau server web adalah sistem yang melayani permintaan suatu *web page*, atau tempat penyimpanan *web page* yang akan dipanggil. *Web page* adalah susunan teks, grafik, suara atau video klip dalam berbagai kombinasi dan berektensikan *.htm atau *.html dan masih banyak lagi lainya, *web page* biasanya berisi *link website* lainya yang dapat diakses dengan mengklik pada *link* yang tersorot, ratusan ribu *website* bertambah setiap hari, dikembangkan oleh individu dan organisasi yang ingin mengumumkan apa yang mereka ketahui, menjual atau yang ingin mereka bagikan ke seluruh dunia.

Server web yang terkenal diantaranya adalah *Apache, Microsoft Internet Information Service (IIS)* dan *Java Server Pages (Sun Java)*. Apache merupakan server web antar-platform, sedangkan IIS hanya dapat beroperasi di sistem operasi Windows.

Server HTTP Apache atau *Server Web/WWW Apache* adalah server web yang dapat dijalankan di banyak sistem operasi (*Unix, BSD, Linux, Microsoft Windows dan Novell Netware* serta platform lainnya) yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web/www ini mengunakan *HTTP*.

Apache memiliki fitur-fitur canggih seperti pesan kesalahan yang dapat dikonfigur, autentikasi berbasis basis data dan lain-lain. Apache juga didukung oleh sejumlah antarmuka pengguna berbasis grafik (*GUI*) yang memungkinkan penanganan server menjadi mudah.

Apache merupakan perangkat lunak sumber terbuka dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation.*

Pada awal mulanya, Apache merupakan perangkat lunak sumber terbuka yang menjadi alternatif dari server web *Netscape* (sekarang dikenal sebagai *Sun Java System Web Server*). Sejak April 1996 Apache menjadi server web terpopuler di internet. Pada Mei 1999, Apache digunakan di 57% dari semua web server di dunia. Pada November 2005 persentase ini naik menjadi 71%. (sumber: *Netcraft Web Server Survey*, November 2005).

Asal mula Apache berasal ketika sebuah server web populer yang dikembangkan awal 1995 yang bernama *NCSA HTTPd* 1.3 memiliki sejumlah perubahan besar terhadap kode sumbernya (patch). Karena banyaknya patch pada perangkat lunak tersebut sehingga disebut sebuah server yang memiliki banyak patch ("*a patchy" server*). Tetapi pada halaman FAQ situs web resminya, disebutkan bahwa "Apache" dipilih untuk menghormati suku asli *Indian Amerika Apache (Indé),* yang dikenal karena keahlian dan strategi perangnya. Versi 2 dari Apache ditulis dari awal tanpa mengandung kode sumber dari *NCSA.*

Apache adalah komponen server web dari paket perangkat lunak *LAMP (Linux, Apache, MySQL, PHP*/*Perl*/bahasa pemrograman *Python*). Menurut statistik dari Netcraft, Apache merupakan server web yang paling banyak digunakan di dunia per 2005 [1]. *Microsoft Internet Information Services (IIS)* merupakan kompetitor utama Apache, diikuti oleh Sun Java Web Server dari Sun Microsystem.