Manajemen Dokumen HTML dengan PHP 5

Didik Dwi Prasetyo
didik@indosql.net
http://didik.indosql.net

Sebagaimana diketahui, dokumen HTML (*HyperText Markup Language*) merupakan dasar terbentuknya aplikasi berbasis web. Pada dasarnya HTML bukanlah sebuah bahasa pemrograman, akan tetapi merupakan semacam bahasa pengkodean. Hal ini disebabkan karena HTML tidak memerlukan kompiler khusus sebagaimana bahasa pemrograman sebenarnya (PHP, ASP, JSP, dan lainnya). Meskipun bukan merupakan bahasa pemrograman, bukan berarti kita bisa sembarang membuat dokumen HTML.
Pembuatan dokumen HTML didasarkan pada aturan-aturan tertentu yang telah disepakati bersama. Oleh sebab itu, ketika Anda lupa sedikit saja dalam membuka atau menutup tag dari elemen, maka bisa berakibat fatal. Mungkin tidak terlalu menjadi masalah ketika kesalahan hanya pada tag biasa semacam paragraf (<P>), akan tetapi bayangkan jika salah menutup tag *table data* (<TD>) atau *table row* (<TR>), dijamin pasti halaman web anda akan berantakan.
Bertolak dari masalah di atas, yakni mengenai penggunaan struktur HTML yang benar, PHP 5 'mengangkut' teknologi khusus dalam melakukan manajemen dokumen HTML. Teknologi yang terpaket dalam fungsi API Tidy ini memungkinkan pemrogram untuk mem-parsing, membersihkan, memproses serta memperbaiki dokumen HTML. Di sini kita akan membahas seperti apa API Tidy ini dan sejauh mana kegunaannya bagi Anda tentunya.

Dokumen HTML

Sebelum melangkah ke penggunaan fungsi API Tidy, ada baiknya kita *review* mengenai dasar dokumen HTML. Anda pasti sudah memahami bahwa dokumen HTML terdiri dari tag dan elemen-elemen. Ada pun ketika *browser* menampilkan hasil dokumen, sebenarnya *browser* menerjemahkan tag-tag menjadi tampilan menarik yang dapat Anda lihat pada *browser* tersebut.
Secara umum dokumen HTML dapat digambarkan sebagai dokumen yang diawali penulisan tag HTML, dengan tanda lebih kecil (<) dan diakhiri atau ditutup dengan tanda lebih besar (>).

Aturan Dasar

Meskipun bukan merupakan bahasa pemrograman, namun dokumen HTML juga memiliki aturan-aturan yang perlu ditaati. Aturan dasar dari dokumen HTML meliputi penulisan tag-tag serta penyimpanan file dokumen.

Beberapa aturan pembuatan dokumen HTML yang perlu Anda ketahui adalah sebagai berikut:

1. Sangat dianjurkan bagi Anda untuk mendefinisikan tipe HTML sebelum memulai pembuatan dokumen HTML. Pendefinisian ini berfungsi untuk menunjukkan bahwa file dokumen yang dibuat adalah dokumen HTML. Definisi yang umum diberikan adalah seperti contoh di bawah ini.

   <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">

   Meskipun sudah diberikan aturan, dalam implementasinya dapat dihitung pemrogram yang mematuhinya. Mengapa demikian? Karena meskipun Anda tidak memberikan definisi tersebut, tetap saja dokumen akan diterjemahkan, dan mungkin hal ini sangat membosankan bagi sebagian pemrogram.

2. HTML akan mengabaikan perbedaan huruf, jadi Anda dapat menuliskan tag-tag dalam bentuk huruf besar semua atau huruf kecil semua. Hal itu tetap akan menghasilkan tampilan yang sama, sebagai contoh perhatikan tag berikut:

   **
**
   **
**

   Sebaiknya ketika Anda membuat suatu dokumen HTML, gunakan huruf yang seragam, misalnya huruf besar semua atau huruf kecil semua. Hal ini akan memudahkan Anda ketika memeriksa ulang program, selain itu juga memudahkan orang lain yang membacanya. Bayangkan jika Anda mencampur adukkan huruf kecil dan huruf besar, tentu jadi kurang enak dilihat, kecuali memang Anda menyukainya.

3. Selain bentuk tag tunggal seperti contoh di atas, ada juga tag yang memiliki pasangan atau penutup, perhatikan contoh berikut:

   <TITLE>…..</TITLE>

   Dalam pembuatan tag yang berpasangan seperti di atas, Anda wajib memberikan tanda ‘/’ (garis miring) yang menandakan penutup pada pasangan tag. Terlihat bahwa <TITLE> merupakan tag awal, sedangkan </TITLE> adalah tag akhir yang berfungsi menutup perintah tersebut.

4. Tanda spasi atau baris baru yang diapit oleh teks akan diabaikan browser, perhatikan contoh berikut:

   ****Menebalkan huruf****

   atau

   ****
   Menebalkan huruf
   ****

   Kedua contoh cara penulisan di atas pada dasarnya akan menghasilkan tampilan yang sama.

5. Dokumen HTML harus disimpan sebagai teks murni dengan menggunakan ekstensi **.html** atau **.htm**.

Struktur HTML

Di luar dari penggunaan definisi yang telah dijelaskan, secara umum dokumen HTML memiliki tiga buah elemen utama yaitu HTML, HEAD, dan BODY (semuanya merupakan tag berpasangan).

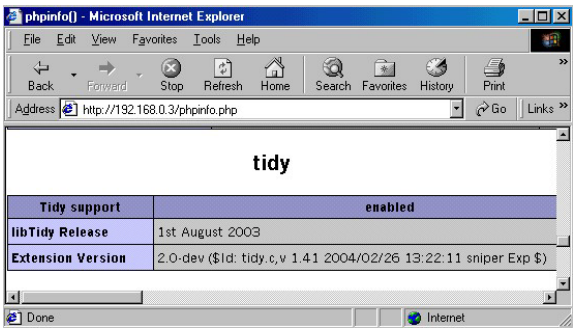
Perlu diketahui, meskipun sudah ada struktur dalam pembuatan dokumen HTML, bukan berarti kita tidak dapat membuat dokumen HTML jika mengabaikan beberapa struktur tersebut. Misalnya Anda membuat dokumen sederhana hanya dengan elemen <HTML> dan kemudian ditutup oleh tag </HTML>. Tetap saja dokumen tersebut akan dianggap sebagai dokumen HTML, sehingga Anda akan dapat melihat hasilnya melalui *browser*.

Konfigurasi Parser Tidy

Langkah utama yang diperlukan sebelum menggunakan fungsi-fungsi Tidy adalah mengaktifkan modul terlebih dahulu. Harap dimaklumi, mengingat cukup banyak modul yang tersedia pada PHP maka untuk mengurangi banyaknya modul-modul yang berjalan, secara normal PHP hanya membatasi beberapa modul umum saja yang sudah aktif begitu Anda menginstal PHP.
Oleh karena modul Tidy tidak termasuk sebagai modul umum, maka kita perlu mengaktifkan modul ini. Untuk lebih jelasnya ikuti tahap-tahap konfigurasi berikut:

1. Apabila web server Anda masih aktif berjalan, sebaiknya matikan terlebih dahulu.
2. Berikutnya cari file **php.ini** yang Anda gunakan, kemudian buka melalui teks editor.
3. Cari modul **php_tidy.dll** dalam kumpulan ekstensi modul, kemudian hilangkan tanda komentar didepannya. Jika Anda menggunakan lingkungan Unix (Linux), baris ini seharusnya menjadi **php_tidy.so**.

4. Jalankan kembali web server Anda, dan periksa melalui fungsi *phpinfo()* untuk memastikan bahwa modul sudah benar-benar aktif.



Gambar 1 Mengaktifkan dukungan modul Tidy

Menampilkan Source HTML

Berbeda dengan program PHP yang tidak dapat dilihat *source code*-nya melalui browser *client*, dokumen HTML memungkinkan *client* untuk membaca *source code*. Jadi ketika Anda membuka halaman web melalui Internet, meskipun dokumen tidak terletak pada penyimpanan lokal, dengan mudah Anda dapat melihat-lihat isinya.
Mengingat aplikasi web umumnya tidak begitu sederhana, sehingga dokumen yang terlihat sangat banyak dan terdiri dari tag-tag yang memusingkan. Misalnya saja Anda ingin melihat *header* dari dokumen untuk mengetahui isi dari style css yang ada pada situs seperti dokumen berikut:

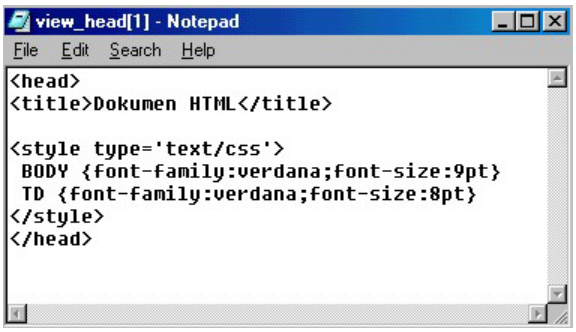
```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title>Dokumen HTML</title>
 <style type='text/css'>
 BODY {font-family:verdana;font-size:9pt}
 TD {font-family:verdana;font-size:8pt}
 </style>
</head>
<body>
```

```
<p>
PHP merupakan pemrograman disisi server, di mana
pemrosesan dilakukan pada komputer server.
</p>

</body>
</html>
```

Untuk dapat mengambil isi dari *header* dokumen di atas, gunakan fungsi *tidy_get_head()* yang akan membantu Anda untuk menampilkan objek dokumen dimulai dari tag <head> sampai tag penutup </head>.

```
<?php
/* view_head.php */

// File dokumen yang akan diperiksa
$html = 'http://192.168.0.3/myphp5/index.html';

$tidy = tidy_parse_file($html);

// Melakukan proses
$head = tidy_get_head($tidy);
echo $head->value;
?>
```

Ketika Anda jalankan pada *browser*, tentu saja halaman yang ditampilkan akan kosong. Mengapa demikian? Meskipun di dalam dokumen terdapat isi paragraf, namun hal itu akan diabaikan. Inilah akibat dari penggunaan fungsi *tidy_get_head()*, di mana hanya akan mengambil isi *header* saja.
Anda dapat melihat bahwa judul dari halaman tersebut sudah sesuia dengan dokumen yang Anda buat. Akan tetapi di mana informasi *header*? Untuk *browser* Internet Explorer, klik menu **View > Source**, maka Anda akan mendapatkan tampilan seperti berikut.



Gambar 2 Mengambil header halaman situs

Sebaliknya ketika Anda hanya ingin menampilkan isi dari body dokumen, gunakan fungsi *tidy_get_body()*.

```
<?php
/* view_head.php */

// File dokumen yang akan diperiksa
$html = 'http://192.168.0.3/myphp5/index.html';

$tidy = tidy_parse_file($html);

// Melakukan proses
$head = tidy_get_body($tidy);
echo $head->value;
?>
```

Perhatikan hasil tampilan yang diberikan, meskipun kita memiliki style yang secara umum akan mengubah font pada body dokumen, namun hal ini tidak akan terjadi. Sebagaimana Anda ketahui, hal ini disebabkan oleh kerja *parser* yang hanya sebatas pada elemen body saja.
Lain halnya ketika Anda menggunakan fungsi *tidy_get_html()*, di mana akan menampilkan dokumen secara lengkap dari awal hingga akhir.

Membuat Struktur HTML

Kali ini kita mencoba untuk membuat struktur HTML secara otomatis yang akan dilakukan oleh *parser*. Anggap saja ketika kita ingin membuat data dalam tabel, namun kurang begitu paham dengan elemen tabel. Cukup rumit bukan? Padahal katakanlah kemampuan kita hanya sebatas mengetahui elemen <td> saja, yakni untuk mendefinisikan data yang kita miliki.
Daripada kita memaksakan diri untuk membuat sesuatu yang tidak kita mengerti, akan lebih bijaksana jika kita menggunakan bantuan fungsi *tidy_get_output()*.
Sekarang coba buat program seperti berikut:

```
<?php
/* struktur_html.php */

$tabel = "<td>Isi Data</td>";
$proses = tidy_parse_string($tabel);

$proses->CleanRepair();

echo tidy_get_output($proses);
?>
```

Setelah Anda jalankan pada *browser*, dengan mudahnya Anda akan diberikan kerangka tabel lengkap dengan data yang Anda berikan (pada view source).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title></title>
</head>
<body>
<table>
<tr>
<td>Isi Data</td>
</tr>
</table>
</body>
</html>
```

Mendeteksi Error pada Dokumen

Meskipun boleh dikatakan bahwa dokumen HTML cukup sederhana, namun tak jarang cukup memusingkan juga. Ini akan semakin terasa ketika kita membuat aplikasi HTML dalam dokumen yang kompleks, di mana hampir semua tag elemen yang ada kita angkut semua. Seberapa teliti kita memberikan tag? misalnya untuk penggunaan tabel.
Salah sedikit saja Anda memberikan tag pembuka serta penutup, bisa-bisa tampilan web akan sekusut pikiran Anda. Nah, dengan memanfaatkan fungsi *parse* dari Tidy, setidaknya akan membantu kita dalam mendeteksi kesalahan pada dokumen. Selanjutnya Anda cukup mencari letak kesalahan melalui pesan yang ditampilkan oleh Tidy. Untuk memahami seperti apa cara kerja fungsi Tidy dalam mendeteksi kesalahan dokumen Anda, buat contoh sederhana dokumen HTML.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
```

```
<head>
<title>Menggunakan Modul Tidy</title>
</head>
<body>
<table border="1" cellpadding="3" summary="">
  <tr>
  <th>Id</th>
  <th>Nama Barang</th>
  <th>Spesifikasi</th>
  <th>Harga</th>
  </tr>

   <tr></td>M-01</td><td>Memori [DDR]</td><td>128 MB</td><td>Rp
200.000</td></tr>
   <tr><td>Ms-7</td><td>Mouse Optik</td><td> - </td><td>Rp 100.000</td></tr>
   <tr><td>C-01</td><td>CD ROM</td><td>52 X</td><td>Rp 250.000</td></tr>

</table>

</body>
</html>
```

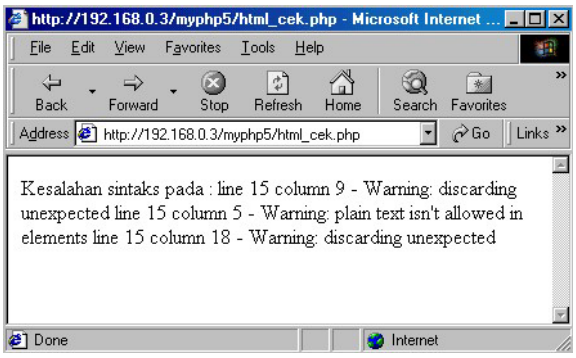
Kalau Anda perhatikan kira-kira program di atas sudah sesuai atau belum? Jika Anda menjawab sesuai berarti Anda kurang jeli. Pada dokumen tersebut, terlihat ada kesalahan yang disebabkan kelebihan menutup tag <td> diawal ID. Sementara anggap saja Anda tidak mengetahui kesalahan tersebut, sehingga tampilannya akan terlihat seperti di bawah ini.

Gambar 3 Pembuatan dokumen HTML yang kurang tepat

Sekarang kita akan membuat program yang akan memeriksa keabsahan dokumen di atas.

```
<?php
/* html_cek.php */

// File dokumen yang diperiksa
$cek = tidy_parse_file('test.html');

// Melakukan proses
$cek->cleanRepair();

// Jika tidak ada kesalahan
if(empty($cek->error_buf)) {
   echo "Sintaks OK";

  } else {
    echo "Kesalahan sintaks pada : ¥n ";
    echo $cek->error_buf;
}
?>
```

Jalankan program yang telah Anda buat di atas, kemudian perhatikan pesan yang ditampilkan.



Gambar 4 Tampilan kesalahan pada dokumen HTML

Masih kurang jelas dengan kesalahan yang ada? Klik menu **View > Source**. Perhatikan pesan kesalahan yang dijelaskan secara lebih detail dalam teks editor.
Setelah mengetahui pesan serta letak kesalahannya, sekarang dengan mudahnya Anda memperbaiki dokumen Anda. Untuk sementara kita memperbaiki secara manual, karena sebenarnya Tidy juga dapat melakukan perbaikan secara otomatis. Bagaimana hal ini dilakukan?

Memperbaiki Dokumen

Salah satu keunggulan paling canggih dari sekian kemampuan fungsi-fungsi Tidy adalah mampu melakukan perbaikan dokumen yang salah. Dalam hal ini kesalahan yang dimaksud adalah kesalahan struktur dokumen. Meskipun belum bisa dikatakan pasti, bahwa kesalahan yang diperbaiki adalah yang Anda inginkan, namun akan sangat membantu sekali. Mengapa dikatakan belum pasti? Karena ini bergantung dari kesalahan yang ditimbulkan.
Sebagai contoh kesalahan dokumen yang Anda buat sebelumnya, ketika diperbaiki maka justru akan menghapus tag <td>. Mengapa demikian? Karena sangat tidak umum jika setelah tag <tr> kemudian ditutup dengan tag </td>, sehingga dengan bijaksana Tidy akan menghapus tag yang tidak berfungsi tersebut.
Untuk menggambarkan kesalahan yang dapat diperbaiki, buatlah contoh dokumen html yang asal, namun jangan mengabaikan elemen utama.

```
<html>
<head>
<title>test dokumen</title>
<head>
<body>

<p>test paragraph<p>

<body>
<html>
```

Tentu sangat tidak wajar bukan? Sekarang coba kita suruh parser Tidy untuk memperbaikinya dengan menggunakan fungsi *tidy_repair_file()*. Berikutnya agar hasil revisi dapat dituliskan ke dalam file, kita memerlukan bantuan fungsi *file_put_contents()*.

```
<?php
/* html_repair.php */

// Target file dokumen
```

```
$file = 'coba.html';

// Proses perbaikan dokumen
$repaired = tidy_repair_file($file);

// Membuat backup file
// dengan ekstensi .bak
rename($file, $file . '.bak');

// Menuliskan string ke dokumen
file_put_contents($file, $repaired);
?>
```

Sengaja pada program di atas kita berikan *backup* dari file dokumen. Ada pun hasil dari perbaikan dokumen di atas akan dapat Anda lihat dalam *source* aplikasi.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<title>test dokumen</title>
</head>
<body>
<p>test paragraph</p>
</body>
</html>
```

Akhirnya, sampai di sini pembahasan mengenai manajemen dokumen HTML dengan PHP 5. Artikel ini diambil dari salah satu bab buku yang ditulis oleh penulis, dengan harapan bermanfaat bagi Anda. Untuk mendapatkan informasi lebih lanjut mengenai API Tidy, kunjungi alamat http://tidy.sf.net/.

Referensi

http://www.php.net/

BIOGRAFI PENULIS

Didik Dwi Prasetya. Lahir di Bojonegoro, 30 September 1979. Menyelesaikan program S1 jurusan Teknik Informatika di Universitas Ahmad Dahlan, Jogjakarta, pada tahun 2004. Saat ini sedang menyelesaikan studi pada jurusan yang sama dengan konsentrasi bidang *Software Engineering* di Institut Teknologi Bandung.
Kompetensi inti adalah pada bidang Web Engineering, Software Engineering, dan Database Management Systems. Kegiatan yang ditekuni sampai sekarang adalah sebagai penulis buku komputer di PT. Elexmedia Komputindo (Gramedia Group).

Penulis juga merupakan salah satu pendiri IndoSQL.net Research Group, komunitas “kecil” pengembang aplikasi Internet dan Database yang berlokasi di Jogjakarta. Selain itu, juga aktif mengelola forum online yang beralamat di http://forum.indosql.net//.

Informasi lebih lanjut tentang penulis ini bisa didapat melalui:
URL: *http://didik.indosql.net*
Email: *didik@indosql.net*