

Membaca dan Menampilkan RSS Feed dengan Perl

Kontribusi: Web Master
Tuesday, 26 September 2006

RSS atau Really Simple Syndication merupakan metode yang sangat populer digunakan untuk menampilkan rangkuman berita atau blog teranyar di website. Situs-situs penyaji berita seperti CNN, BBC, Reuters, dan situs-situs blog ternama lainnya menyediakan RSS feed secara bebas. RSS disusun dalam format XML yang terdiri dari beberapa elemen seperti judul, URL, deskripsi, tanggal publikasi, dan beberapa elamen lainnya. Sebuah RSS feed biasanya ditandai dengan icon . Tulisan ini membahas secara gamblang bagaimana RSS feed dapat dibaca dan ditampilkan di website menggunakan PERL.

Sebelum kita membahas contoh program dari tulisan ini, perlu diketahui bahwa CPAN (Comprehensive PERL Archive Network) menyediakan cukup banyak modul dan subroutine yang dapat secara gratis digunakan. Sebagai contoh, modul LWP::Simple, yang pada contoh program di bawah ini digunakan untuk menyalin (men-download) halaman website. Kumpulan RSS feed Disimpan Dalam File Teks Pada contoh ini, kumpulan RSS feed dikumpulkan dalam sebuah file. File ini dinamakan rss_feed.lst. Setiap baris merepresentasikan sebuah RSS feed. Kolom pertama merupakan URL dari RSS feed yang dimaksud, dan kolom kedua menyatakan jumlah item dari RSS feed tersebut yang akan ditampilkan di website. Antar kolom dipisahkan dengan spasi. Berikut sekilas gambaran data yang disimpan dalam rss_feed.lst. http://www.linuxtoday.com/backend/biglt.rss 10http://rss.cnn.com/rss/cnn_tech.rss 5 Baris pertama dari contoh di atas merupakan RSS feed dari linuxtoday.com dan 10 item dari sumber ini akan ditampilkan, ditandai dengan angka 10 pada kolom kedua baris tersebut. Sedangkan baris kedua merupakan RSS feed dari situs CNN tentang masalah teknologi dan untuk yang satu ini, hanya 5 item yang akan ditampilkan.

Modul ParseRSS Modul ini berfungsi mem-parse elemen-elemen RSS feed yang disusun dalam format XML. Modul ini nantinya akan dipanggil oleh program news.cgi, sebuah skrip cgi yang akan menampilkan item-item RSS feed di browser. #!/usr/local/bin/perl -w

```
package module::ParseRSS;
use strict;
```

```
sub new{
    my $pkg = shift;
    my $res = shift;
    my $channel;
    my $source;
    my $sourceURL;
    my $titlList = "";
    my $dateList = "";
    my $linkList = "";
    my $descList = "";
    if($res =~ /.?<channel>.*?<title>(.*?)</title>.*?<link>(.*?)</link>(.*?)</channel>/sg){
        $source = $1;
        $sourceURL = $2;
        $channel = $3;
    }
    if(defined $channel){
        chomp($channel);
        $channel =~ s/.?<Vimage>//sg;
        my @items = split('<Vitem>', $channel);
        foreach my $line(@items){
            chomp($line);
            if($line =~ /<title>(.*?)</title>.*?<pubDate>(.*?)</pubDate>.*?<link>(.*?)</link>.*?<description>(.*?)</description>/s){
                $titlList .= $1 . '_DEL_';
                $dateList .= $2 . '_DEL_';
                $linkList .= $3 . '_DEL_';
                $descList .= $4 . '_DEL_';
            }
        }
    }
    my $private = {
        titlList => $titlList,
        dateList => $dateList,
        linkList => $linkList,
        descList => $descList,
        source => $source,
        sourceURL => $sourceURL,
```

```

};
bless $private, $pkg;
}

sub getTitle{
    my $obj = shift;
    return $obj->{titlList};
}

sub getDate{
    my $obj = shift;
    return $obj->{dateList};
}

sub getLink{
    my $obj = shift;
    return $obj->{linkList};
}

sub getDesc{
    my $obj = shift;
    return $obj->{descList};
}

sub getSource{
    my $obj = shift;
    return $obj->{source};
}

sub getSourceURL{
    my $obj = shift;
    return $obj->{sourceURL};
}

```

return 1; Pendeklarasian sebuah modul dilakukan di bagian awal dengan menggunakan keyword package seperti contoh di bawah ini. package module::ParseRSS; Modul ini disimpan dalam subdirektori module dengan nama ParseRSS.pm. Subroutine new adalah subroutine utama dalam modul ini. Halaman website yang didownload menggunakan LWP::Simple dikirim ke dalam subroutine ini (pass by value) dan disimpan dalam variabel \$res. Selanjutnya, baris per baris dari variabel \$res ini di-parse dan item judul, tanggal publikasi, URL, dan deskripsi diambil nilainya. Pada setiap iterasi, item judul, tanggal, descripsi, dan URL disimpan dalam skalar yang dipisahkan dengan simbol '_DEL_'. foreach my \$line(@items){
 chomp(\$line);
 if(\$line =~ /<title>(.*?)</title>.*?<pubDate>(.*?)</pubDate>.*?<link>(.*?)</link>.*?<description>(.*?)</description>/s){
 \$titlList .= \$1 . '_DEL_';
 \$dateList .= \$2 . '_DEL_';
 \$linkList .= \$3 . '_DEL_';
 \$descList .= \$4 . '_DEL_';
 }
}

Kumpulan dari item-item ini dapat dengan mudah diambil dan dipisah-pisahkan kembali menggunakan fungsi split (lihat penggunaan fungsi split pada skrip news.cgi berikut ini).

Bagian Utama (news.cgi) Pada bagian ini, module ParseSSH dipanggil. Pemanggilan modul ini dilakukan menggunakan keyword use. Ada dua build-in modul yang digunakan dalam skrip ini: modul CGI dan LWP::Simple. Seperti yang telah penulis sebutkan sebelumnya, semua modul-modul ini dapat didownload dari CPAN website.

```
#!/usr/bin/perl -w
```

```

use strict;
use CGI;
use LWP::Simple;
use module::ParseRSS;

my $cgi = new CGI;
my $seed_file = "data/rss_seed.lst";

print $cgi->header(-type=>'text/html',

```

```

-charset=>'utf8');

print $cgi->start_html({-title=>"Keudekupi.com News",
                      -style=>{-src=>'../styles/news.css'}});

my @seeds = ();
get_seeds($seed_file, \@seeds);
print_results(\@seeds);

print $cgi->end_html;

sub get_seeds{
  my $seed_file = shift;
  my $seed_ref = shift;
  open(SEED, $seed_file) || return;
  while(<SEED>){
    next if /^\s*$/;
    next if /^#+/;
    push(@$seed_ref, $_);
  }
  close(SEED);
}

sub print_results{
  my $seed_ref = shift;
  my @seeds = @$seed_ref;
  print $cgi->div({-class=>'news_container'});
  foreach my $seed(@seeds){
    my @entry = split('\s+', $seed);
    my $result = get($entry[0]);
    my $displayNum = $entry[1];
    my $obj = new module::ParseRSS($result);
    my @title = split("_DEL_", $obj->getTitle);
    my @date = split("_DEL_", $obj->getDate);
    my @link = split("_DEL_", $obj->getLink);
    my @desc = split("_DEL_", $obj->getDesc);
    print $cgi->div({-class=>'news'}); for my $i(0..$displayNum-1){
      print $cgi->div({-class=>'teubai'});
      print $cgi->a({-href=>"$link[$i]", "title[$i]"} . "<br>\n";
      print "</div>\n";
      print "$desc[$i]<br>\n";
      my $sourceURL = $obj->getSourceURL;
      my $source = $obj->getSource;
      print $cgi->div({-class=>'singer'});
      print "Source: " . $cgi->a({-href=>"$sourceURL", "$source"} . ", $date[$i]<br>\n";
      print "</div>\n";
      print "<br>\n";
    }
    print "</div>\n";
  }
  print "</div>\n";
}

```

Subroutine `get_seeds` membaca file `rss_feed.lst`. Tiap baris yang ada dalam file tersebut disimpan dalam array menggunakan fungsi `push`, seperti yang diperlihatkan pada pernyataan berikut ini: `push(@$seed_ref, $_)`; Setelah file dibaca, hasil akan ditampilkan dilayar satu per satu. Semua format tampilan disusun di subroutine `print_results`. Pemanggilan module `ParseRSS` dilakukan di subroutine ini. `my $obj = new module::ParseRSS($result)`; Pemisahan item-item dilakukan dengan fungsi `split`. Setiap item yang dipisahkan diantara simbol `'_DEL_'` kemudian disimpan dalam array. Dalam PERL, sebuah array ditandai dengan simbol `@`.

```

my @title = split("_DEL_", $obj->getTitle);
my @date = split("_DEL_", $obj->getDate);
my @link = split("_DEL_", $obj->getLink);
my @desc = split("_DEL_", $obj->getDesc);

```

Pemanggilan fungsi-fungsi dalam modul `ParseRSS` dilakukan dengan menggunakan variabel `$obj` diikuti tanda `->` dan

nama fungsi yang dipanggil.

Penutup Cara yang paling mudah memahami suatu program adalah dengan menjalankan program itu secara langsung. Oleh karena itu, penulis menyarankan agar program di atas disimpan dengan nama yang sesuai dan jalankan. Perhatikan hasilnya agar apa yang telah dibahas dalam tulisan ini benar-benar dapat dipahami. Untuk diskusi lebih lanjut, penulis dapat dihubungi melalui email: taufik.abidin@keudekupi.com.