

# Proteksi Direktori di Web dengan .htaccess

Bagi pengguna web, terkadang kita ingin memproteksi sebuah direktori agar hanya dapat diakses oleh *user* tertentu, dengan *password* tertentu juga. Berbagai cara tentu bisa dilakukan, namun apabila menggunakan web server Apache, kita bisa memanfaatkan file **.htaccess**. Mudah dan cepat.

Ketika bekerja dengan web, sebagian besar tugas yang dilakukan adalah menampilkan konten dalam format *hypertext*, yang memiliki *layout* yang menarik serta berbagai fitur interaktif lainnya.

Namun, apabila kita bicara soal web, kita juga akan bicara soal berbagi. Berbagi informasi ataupun berbagi berbagai hal lainnya. Berbagi informasi bisa dilakukan dalam banyak hal. Salah satunya, seperti yang disebutkan di atas, dengan menampilkan konten dan informasi dalam format *hypertext*. Sementara, untuk berbagi file misalnya, banyak pengguna web yang lebih senang untuk memanfaatkan fasilitas *directory listing* yang dimiliki oleh berbagai web server populer di dunia ini.

Fasilitas *directory listing* berguna untuk menampilkan isi sebuah direktori melalui web. Dengan demikian, user akan dapat mendownload file-file yang ditampilkan dengan mudah, mirip dengan user tersebut menggunakan file manager seperti Konqueror ataupun Nautilus yang mengaktifkan *detailed/list view*. Namun, karena aplikasi yang digunakan adalah web browser dan *directory listing* hanya bermaksud untuk menampilkan isi direktori apa adanya, berbagai fitur file manager seperti *select all*, *copy*, *cut*, *paste* atau *rename* tidak akan tersedia. Hanya fitur-fitur standar (masuk dan keluar direktori, *sort*, *download*) yang disediakan.

Untuk file yang ingin dibagi ke publik, hal ini tentu sudah sangat memuaskan. Tidak ada satu kode HTML pun yang perlu kita buat. Apalagi kalau harus sampai menulis berbagai script. Cukup melakukan konfigurasi web server.

Sayangnya, terkadang, kita memiliki file-file di dalam direktori tertentu yang hanya boleh dibagi kepada pihak tertentu. Jadi, dari sepuluh direktori yang kita *share*, ada sebuah direktori penting yang tidak boleh diakses siapa saja. Untuk mengaksesnya, user harus memasukkan *username* dan *password*.

Sampai sejauh ini, kita masih bisa memanfaatkan fasilitas *protected directory* yang telah tersedia pada berbagai web server populer. Apache adalah salah satunya. Pada Apache, proteksi ini dimungkinkan dengan memanfaatkan file **.htaccess**. Tutorial ini akan membahas cara-cara untuk membuat *protected directory* pada web server Apache memanfaatkan **.htaccess**.

## Pengenalan .htaccess

File **.htaccess** adalah sebuah file yang umumnya bernama **.htaccess** (perhatikan tanda titik di depan **htaccess**) yang berisikan aturan-aturan atau opsi-opsi yang dikenakan pada direktori yang mengandung file ini. File ini bisa pula berisikan aturan user siapa saja, atau group siapa saja yang boleh mengakses direktori yang mengandung file ini. File **.htaccess** merupakan sebuah file teks yang bisa dibuat dengan *text editor* biasa.

File **.htaccess** sering kali disebutkan sebagai **htaccess** saja, atau seringkali disebutkan juga sebagai *distributed configuration file*. Disebut *distributed configuration file* karena file **.htaccess** bisa di-*copy*-kan ke direktori mana saja dan akan langsung berdampak pada direktori tersebut ketika diakses melalui web browser.

Secara *default*, nama file yang digunakan adalah **.htaccess**. Namun, apabila nama

file tersebut tidak disukai, pengguna selalu bisa mempergunakan nama file lain dengan mengubah *directive AccessFileName*. Sebagai contoh:

```
AccessFileName .config
```

Sintaks yang digunakan dalam file **.htaccess** adalah sama seperti halnya dengan file konfigurasi utama Apache.

Penggunaan file **.htaccess** memiliki sejumlah kelebihan, seperti yang disebutkan sebelumnya. Kita dapat lebih leluasa dalam berbagi file tanpa harus repot-repot membangun script sendiri ataupun dengan cara-cara lain yang lebih rumit. Namun, semua kelebihan ini harus dibayar dengan:

- Bertambahnya *resource* yang dibutuhkan oleh web server karena web server akan mencari file **.htaccess** (atau file lain yang diubah melalui *directive AccessFileName*) di setiap direktori. Bahkan, untuk setiap request ke file tertentu, file **.htaccess** akan diproses. Semua ini ditambah lagi dengan web server akan mencari ke direktori-direktori di level atas untuk sepenuhnya menjamin agar penggunaan **.htaccess** dilakukan dengan benar. Dengan demikian, ketika kita meng-*enable* fasilitas **.htaccess** ini, digunakan ataupun tidak, akan berpengaruh pada performa dan *resource* yang dibutuhkan.
- Keamanan. Dengan penggunaan **.htaccess**, konfigurasi server tidak sepenuhnya lagi dikontrol lewat konfigurasi utama web server. Celah keamanan bisa datang dari dalam (*user* yang nakal), ataupun dari luar (*file konfigurasi* yang salah, bisa berefek pada gangguan dari luar).

- Kerepotan untuk me-maintain database password. Penggunaan .htaccess memungkinkan user untuk menggunakan berbagai database password di berbagai lokasi.

Tulisan ini akan mengasumsikan nama file yang digunakan adalah tetap .htaccess. Pada bagian berikutnya, kita akan melihat apa yang harus diatur pada konfigurasi Apache agar .htaccess bisa diterapkan.

## Konfigurasi pada Apache

Apabila Anda adalah administrator web server, maka lakukanlah langkah-langkah berikut ini:

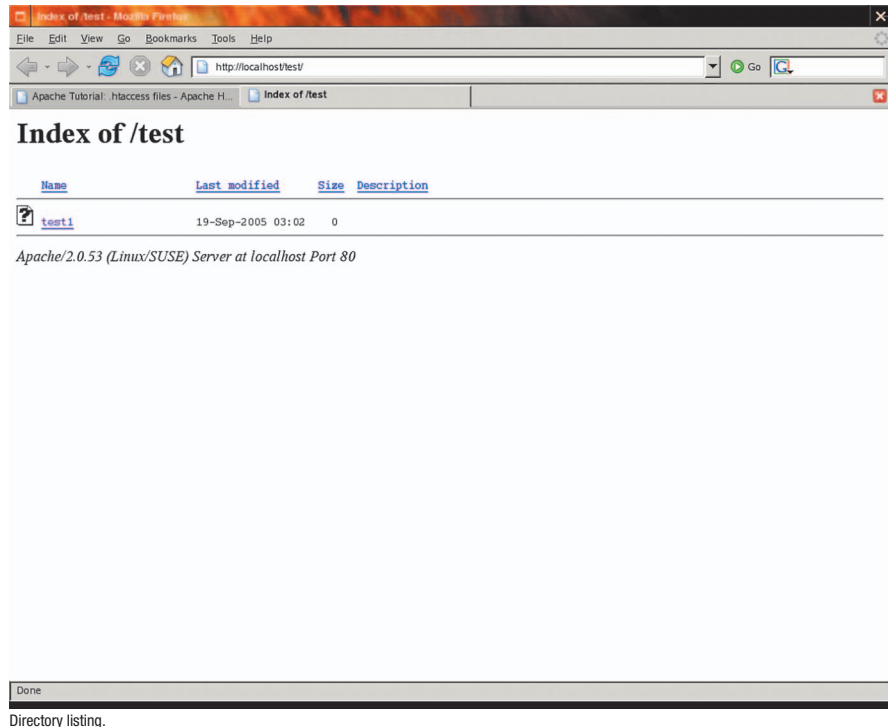
- Bukalah konfigurasi web server. Pada beberapa distro, nama file konfigurasi adalah httpd.conf. Pada distro SUSE, editlah file /etc/apache2/default-server.conf.
- Pada bagian untuk opsi direktori DocumentRoot, berikan opsi AllowOverride AuthConfig. Di sistem default, opsi untuk AllowOverride mungkin akan bernilai None.

Berikut ini adalah contoh konfigurasi DocumentRoot penulis:

```
<Directory "/srv/www/htdocs">
    Options Indexes
    MultiViews
    AllowOverride AuthConfig
    Order allow,deny
    Allow from all
</Directory>
```

Apabila Anda tidak memiliki akses pada konfigurasi utama (misal, Anda berada di jaringan besar dimana Anda bukan administrator Apache), maka mintalah kepada administrator Anda untuk mengaktifkan penggunaan .htaccess. Setelah diaktifkan, Anda langsung bisa membuat file .htaccess.

Apabila Anda menggunakan jasa web hosting, maka pastikan web hosting yang Anda gunakan mendukung fasilitas ini. Penulis menggunakan web hosting Master Web Network (masterwebnet.com), dimana fasilitas .htaccess telah didukung. Apabila .htaccess memang telah didukung, Anda tidak perlu melakukan konfigurasi apapun di web server. File .htaccess bisa langsung dibuat.



## Pembuatan file .htaccess sederhana

Sampai pada bagian ini, kita mengasumsikan .htaccess telah didukung di web server. Selanjutnya, kita akan membuat sebuah direktori test di bawah DocumentRoot.

Di sistem yang penulis gunakan, DocumentRoot terletak pada /srv/www/htdocs. Direktori test yang akan kita buat tersebut akan terletak di /srv/www/htdocs/test.

Cobalah untuk mengakses direktori test. Opsi yang kita berikan pada DocumentRoot dan turunannya seharusnya akan memungkinkan isi direktori test ditampilkan apabila diakses (directory listing diaktifkan).

```
http://localhost/test/
```

Setelah berhasil, kita siap untuk bekerja dengan .htaccess. Untuk lebih baiknya, kita akan membuat sebuah file dengan nama test1 di dalam direktori tersebut:

```
$ touch test1
$ ls
total 0
drwxr-xr-x 2 nop users 72
2005-09-19 03:02 ./
drwxr-xr-x 4 root root 96
2005-09-19 02:39 ../
-rw-r--r-- 1 nop users 0
2005-09-19 03:02 test1
```

Selanjutnya, kita akan membuat file .htaccess sederhana di dalam direktori test dengan isi file sebagai berikut:

```
AuthType Basic
AuthName "Test .htaccess"
Require user test
```

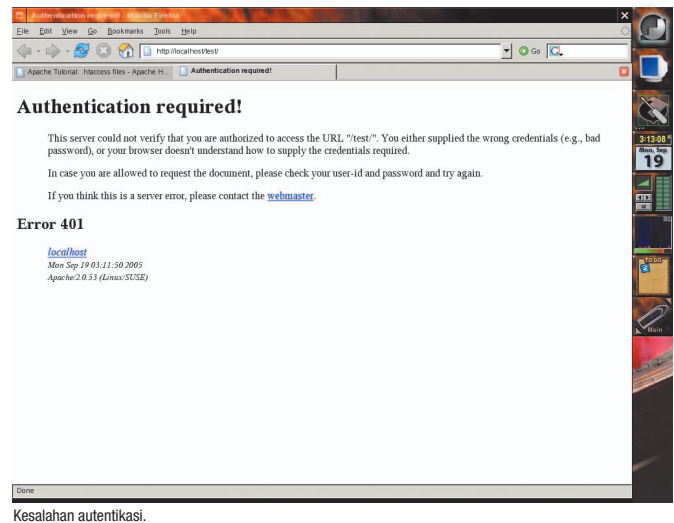
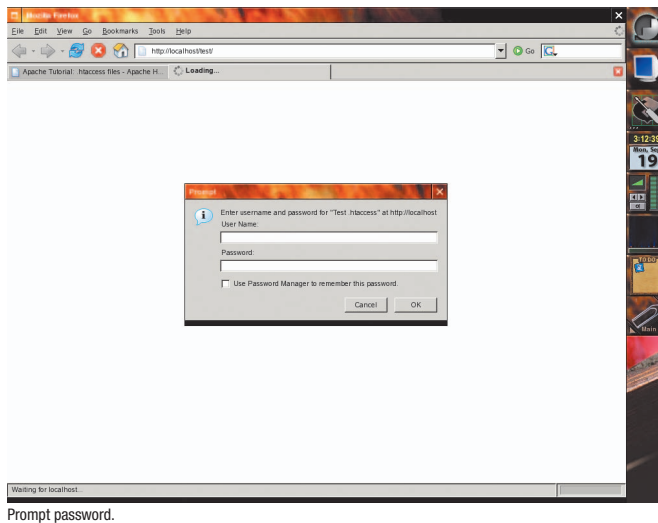
Apabila kita mengakses kembali URL http://localhost/test/ tersebut, maka sebuah kotak dialog yang meminta kita untuk memasukkan user name dan password akan ditampilkan. Pastikan juga tertulis Test .htaccess di kotak dialog tersebut. Browser yang berbeda akan menampilkan kotak dialog yang berbeda.

Saat ini, kita belum memiliki informasi user dan password. Oleh karena itu, batalkanlah kunjungan kita ke localhost/test/. Apa yang kita lakukan sebelumnya hanyalah percobaan untuk melihat apakah .htaccess telah bekerja atau belum.

Apabila kunjungan dibatalkan, maka akan muncul kesalahan 401 (Authentication Required). Hal ini wajar karena web server mengharapkan proses autentikasi dilakukan dengan benar.

Sekarang, mari kita lihat file .htaccess yang kita buat sebelumnya:

- AuthType. Directive AuthType menentukan tipe autentikasi untuk direktori. Saat ini, hanya Basic dan Digest yang



didukung. Directive ini dikombinasikan dengan AuthName, Require dan AuthUserFile, ataupun AuthGroupFile.

- AuthName. Directive ini akan mengatur nama otorisasi untuk direktori. Nama yang digunakan di sini dimaksudkan untuk memberikan deskripsi kepada pengunjung. Directive ini dikombinasikan dengan AuthType, Require dan AuthUserFile ataupun AuthGroupFile.
- Require. Directive ini akan menentukan siapa saja yang diperbolehkan untuk mengakses resource. Directive ini dikombinasikan dengan AuthName, AuthType dan AuthUserFile ataupun AuthGroupFile. Berikut ini adalah bentuk penggunaan Require.

```
Require user userid [userid]
...
```

Hanya nama yang disebutkan yang bisa mengakses resource ini.

```
Require group group-name
[group-name] ...
```

Hanya group yang disebutkan yang bisa mengakses resource ini.

```
Require valid-user
```

Semua user yang valid bisa mengakses resource ini.

Setelah ini, kita akan melanjutkan ke pengaturan database user.

## Pengaturan database user

Database user dapat dibuat dengan program

htpasswd2. Apabila kita login sebagai user biasa, maka program ini mungkin tidak terdapat dalam PATH. Umumnya, htpasswd2 diinstall di /usr/sbin/htpasswd2. Walaupun htpasswd2 diinstall di /usr/sbin, namun user biasa tetap dapat menggunakannya.

Program htpasswd2 adalah program yang dapat digunakan untuk mengatur file database password yang dapat digunakan untuk autentikasi basic. Berikut ini adalah cara penggunaannya:

```
htpasswd [ -c ] [ -m ] [
-D ] passwdfile username
htpasswd -b [ -c ] [ -m |
-d | -p | -s ] [ -D ] passwdfile
username password
htpasswd -n [ -m | -d |
-s | -p ] username
htpasswd -nb [ -m | -d |
-s | -p ] username password
```

Penjelasan opsi:

- opsi -c. Opsi ini digunakan untuk membuat file baru. Apabila file telah tersedia, maka isinya akan ditulis ulang. Berhati-hatilah menggunakan opsi ini.
- Opsi -m. Mempergunakan enkripsi MD5.
- Opsi -D. Menghapus user apabila telah terdaftar di database password yang dispesifikasikan.
- Opsi -b. Mempergunakan mode batch. Penggunaan mode batch akan memungkinkan pemberian password misalnya melalui command line tanpa harus meminta kepada user.
- Opsi -d. Mempergunakan enkripsi melalui crypt()

- Opsi -p. Mempergunakan plain password
- Opsi -s. Mempergunakan enkripsi SHA
- Opsi -n. Hanya menampilkan ke standar output, tidak meng-update database password.

Sebelum kita melanjutkan, kita akan melihat lokasi penyimpanan database password. Berikut ini adalah beberapa hal yang perlu diperhatikan seputar lokasi database password:

- Database password bisa diletakkan di mana saja di sistem ataupun di bawah tree dokumen yang bisa diakses oleh web server.
- Namun, usahakan untuk menyimpan database password diluar tree dokumen yang bisa diakses oleh web server.
- Satu hal yang pasti, jangan sampai pernah menyimpan database password di direktori yang diprotek karena user bisa mendownload database password tersebut. Jangan simpan database password di direktori yang bisa diakses melalui web browser.

Dalam contoh kali ini, kita akan menyimpan database password di /tmp. Sesuai dengan lokasi yang Anda inginkan. Berikut ini, kita akan membuat sebuah user dengan nama test (password: test). Sementara, untuk database password, kita akan memberikan nama test.passwd. Nama file database password bisa disesuaikan dengan keinginan. Berikut ini adalah perintah yang dapat digunakan:

```
$ /usr/sbin/htpasswd2 -c /tmp/
test.passwd test
New password:
Re-type new password:
Adding password for user test
```

Setelah perintah diberikan, kita akan memiliki /tmp/test.passwd yang isinya adalah sebagai berikut pada komputer penulis:

```
$ cat /tmp/test.passwd
test:VdqfKTx9uRoAI
```

Setelah database password kita miliki, berikut ini kita akan melengkapi .htaccess kita agar bisa berfungsi sepenuhnya.

### Melengkapi .htaccess

Di .htaccess kita sebelumnya, kita sudah memiliki beberapa entri. Kita sudah menyebutkan bahwa kita akan mempergunakan autentikasi basic. Kita juga sudah menyebutkan deskripsi resource. Dan, yang paling penting, kita juga telah membatasi agar hanya user test yang diizinkan masuk ke sistem.

Di bagian ini, kita akan menambahkan satu directive lagi, yaitu AuthUserFile. Directive AuthUserFile ini akan mengacu ke /tmp/test.passwd yang kita buat sebelumnya.

Berikut ini .htaccess kita selengkapnya:

```
AuthType Basic
AuthName "Test .htaccess"
AuthUserFile "/tmp/test.passwd"
Require user test
```

Ketika menyebutkan nilai untuk directive AuthUserFile, kita bisa menggunakan path absolut ataupun path relatif. Dalam contoh, kita mempergunakan path absolut. Apabila path relatif diberikan, maka file akan diperlakukan relatif dari directive ServerRoot.

Sampai di sini, kita telah memiliki sebuah .htaccess yang bekerja dengan baik. Cobalah.

### Bagaimana kalau .htaccess tidak bekerja?

Berikut ini adalah beberapa langkah yang bisa dilakukan apabila .htaccess tidak bekerja sesuai keinginan:

- Pastikan Anda telah memiliki directive AllowOverride AuthConfig, bukan AllowOverride None pada direktori yang ingin menggunakan .htaccess.
- Apabila Anda yakin telah mengatur point sebelumnya dengan benar namun .htaccess tetap tidak bekerja, cobalah untuk memberikan sintaks yang salah pada file .htaccess. Apabila tidak ada pesan kesalahan, maka berarti .htaccess Anda tidak dipedulikan. Periksa kembali pengaturan direktori yang ingin diberikan .htaccess. Apabila muncul pesan kesalahan, maka mungkin terjadi kesalahan pada .htaccess Anda. Periksa kembali.
- Selalu pastikan bahwa user, group dan file database password ataupun database group sudah diatur dengan benar sebelumnya.

Sampai di sini dulu pembahasan kita tentang penggunaan .htaccess. Selamat mencoba dan sukses! 🙌

**Noprianto** ([noprianto@infolinux.co.id](mailto:noprianto@infolinux.co.id))

## Advertorial

## Multimedia and Gaming: The Next Killer Industry?



Industri animasi dunia bernilai sekitar US\$500 miliar. Ini merupakan nilai yang amat besar bila dibandingkan dengan nilai total ekspor Indonesia tahun 2004 yang berjumlah US\$ 69.71 miliar. Bayangkan bila Indonesia bisa mengambil peran dalam industri tersebut, berapa banyak devisa yang bisa diperoleh. Tentunya hal ini bisa meningkatkan kinerja perekonomian nasional yang semakin lesu karena harga minyak dunia yang semakin melambung dan industri yang jalan ditempat.

Kecenderungan industri animasi di USA, Korea dan Jepang sekarang adalah offshore production dan outsourcing. Ini disebabkan meroketnya biaya produksi di negara-negara tersebut, yang mencapai US\$5,000 sampai US\$10,000 per menit. Sebagai perbandingan, ongkos produksi film animasi 3-D di India adalah separuh dari ongkos produksi di USA, dan lebih murah lagi di Cina. Tidak mengherankan bila studio-studio animasi menjamur di India dan Cina, dan sebagian besar produksinya merupakan proyek outsourcing atau relokasi dari USA, Korea maupun Jepang.

Melihat perkembangan yang berlangsung dari tersebut, Jurusan Computer Science, BiNus International, ingin ikut mendorong berkembangnya industri animasi di dalam negeri, dengan menyelenggarakan kompetisi "Multimedia and Computer Graphics Challenge 2005". Kompetisi ini bertujuan mendorong bertumbuhnya komunitas multimedia dan animasi, baik pengembang maupun industri, yang diharapkan akan turut mempercepat perkembangan industri animasi di tanah air. Kompetisi ini akan dilaksanakan dalam dua tahap, yang berlangsung dari tanggal 26 Oktober 2005 sampai dengan tanggal 25 Februari 2006. Tahap pertama berupa kompetisi on-line sedangkan tahap kedua adalah kompetisi on-site. Murdoch University, MAHAKA dan 3DsMax merupakan sponsor utama acara ini. Antusiasme masyarakat dapat dilihat dari jumlah dan asal sekolah peserta. Lebih dari 300 peserta dari 15 universitas dan 20 sekolah, berasal dari 10 kota di Indonesia, akan mengikuti kompetisi ini.