

Tutorial Instalasi Django Menggunakan VirtualEnv*

Kholid Fuadi

@sopier

4 Agustus 2012

Daftar Isi

1	Instalasi Virtualenv	2
2	Menggunakan Virtualenv	2
3	Menggunakan Virtualenvwrapper	2
4	Instalasi Django	3
5	Konfigurasi Django	4
6	Mengaktifkan Modul Admin	5
7	Creating Your First App	7

* Artikel ini ditulis menggunakan L^AT_EX dan VIM

1 Instalasi Virtualenv

Apa itu Virtualenv¹? Virtualenv memungkinkan Anda bekerja dengan environment khusus, terbebas dari paket/modul yang sudah terinstall secara *system wide* di komputer Anda.

Misal Anda memiliki paket Django versi 1.3 bawaan dari OS Ubuntu, kemudian Anda ingin mencoba instalasi Django 1.4 tanpa harus melakukan upgrade paket bawaan, Anda dapat membuat sebuah *virtual environment* tersendiri, kemudian melakukan instalasi Django versi 1.4.

Cara install virtualenv:

```
$ sudo pip install virtualenv
```

2 Menggunakan Virtualenv

Anda dapat memulai virtualenv dengan cara:

```
$ virtualenv testing --no-site-packages
$ source testing/bin/activate
(testing)$
# kalau sudah selesai
$ deactivate
```

3 Menggunakan Virtualenvwrapper

Salah satu kekurangan dari virtualenv adalah ketika kita bekerja dengan banyak *virtual environment*, kita menjadi kesulitan dalam menangani *virtual environment* satu persatu, bagaimana solusi untuk mempermudah manajemen *virtual environment* tersebut? Salah satunya adalah menggunakan paket *virtualenvwrapper*.

Berikut cara instalasi *virtualenvwrapper*:

```
$ sudo pip install virtualenvwrapper
```

Setelah itu lakukan konfigurasi pada file `.bashrc` yang terletak di folder `home`

```
$ vim ~/.bashrc
```

Tambahkan dua baris berikut:

```
export WORKON_HOME=~/.Project
source '/usr/local/bin/virtualenvwrapper.sh'
# atau
# source 'which virtualenvwrapper.sh'
```

Simpan dan keluar, setelah itu jalankan perintah berikut:

¹<http://www.virtualenv.org>

```
$ source .bashrc
```

Setelah itu, Anda seharusnya dapat menjalankan perintah `mkvirtualenv` dari `prompt`.

```
$ cd Proyek
$ mkvirtualenv myproject
$ workon myproject
(myproject)$ pip freeze
```

Output dari perintah `pip freeze` adalah paket yang tersedia dalam proyek Anda sekarang. Kelebihan dari `Virtualenvwrapper` adalah kemudahan dalam manajemen *virtual environment* yang Anda miliki, salah satunya adalah perintah `workon` untuk mengetahui *virtualenv* mana saja yang ada.

```
$ workon
django
mygame
```

Anda dapat menghapus *virtualenv* tersebut dengan perintah `rmvirtualenv`

```
$ rmvirtualenv django
Removing django...
$ rmvirtualenv mygame
Removing mygame...
```

Nah, sekarang Anda sudah tahu bagaimana cara *setting* lingkungan untuk melakukan *development* sebuah aplikasi. Mari kembali ke proyek Django kita. Jika Anda ingin install Django versi terbaru (1.4.1), berikut langkah-langkahnya:

4 Instalasi Django

Membuat `virtualenv` untuk proyek django Anda

```
$ cd Proyek
$ mkvirtualenv django
Installing setuptools.....done.
Installing pip.....done.
virtualenvwrapper.user_scripts creating /home/banthink/Project/django/bin/predeactivate
virtualenvwrapper.user_scripts creating /home/banthink/Project/django/bin/postdeactivate
virtualenvwrapper.user_scripts creating /home/banthink/Project/django/bin/preactivate
virtualenvwrapper.user_scripts creating /home/banthink/Project/django/bin/postactivate
virtualenvwrapper.user_scripts creating /home/banthink/Project/django/bin/get_env_details
(django)$ pip freeze
argparse==1.2.1
wsgiref==0.1.2
```

Memulai install django di `virtualenv`, pastikan Django sudah Anda download dan berada pada direktori *virtualenv* anda (dalam contoh ini berada pada direktori `testing`).

```
(django)$ cd Proyek/django
(django)$ tar zxvf Django-1.4.1.tar.gz
(django)$ cd Django-1.4.1
(django)$ python setup.py install
(django)$ pip freeze
Django==1.4.1
argparse==1.2.1
wsgiref==0.1.2
(django)$ rm Django-1.4.1.tar.gz
(django)$ rm -r Django-1.4.1
```

Sampai saat ini, struktur direktori Anda kurang lebih akan seperti ini

```
.
├── bin
├── include
├── lib
└── local
```

Setelah Django terinstall, Anda dapat memulai proyek baru dengan cara:

```
(django)$ django-admin.py startproject mysite
```

Dan berikut adalah struktur direktori dari `mysite`

```
mysite
├── manage.py
├── mysite
│   ├── __init__.py
│   ├── settings.py
│   ├── urls.py
│   └── wsgi.py
```

Untuk memastikan bahwa django sudah terinstall dengan baik, jalankan perintah berikut:

```
$ cd mysite
$ ./manage.py runserver
```

setelah itu buka browser Anda dan ketikkan pada URL:

http://127.0.0.1:8000/

Jika ada *greeting* **It worked!** berarti django sudah terinstall dengan baik pada *virtual environment* Anda. Selamat!

5 Konfigurasi Django

Hal pertama yang harus di konfigurasi ketika pertama kali install Django adalah file `settings.py`, yakni setting database. Untuk contoh kali ini akan saya gunakan engine *sqlite3*.

Cari bagian berikut pada file `settings.py`:

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.', # Add 'postgresql_psy...
        'NAME': '',                      # Or path to database...
        'USER': '',                      # Not used with sqlite3.
        'PASSWORD': '',                  # Not used with sqlite3.
        'HOST': '',                      # Set to empty string...
        'PORT': '',                      # Set to empty string...
    }
}

```

Edit bagian **ENGINE** (*engine database* yang ingin Anda gunakan) dan **NAME** (nama *database*). Jika Anda menggunakan sqlite3, gunakan *full path* ke direktori file *database* Anda. Misal saya ingin menyimpannya di direktori *mysite*, dengan nama *mysite.db*, maka berikut konfigurasi saya:

```

'ENGINE': 'django.db.backends.sqlite3',
'NAME': '/home/banthink/django/mysite/mysite.db',

```

Setelah itu jalankan perintah `syncdb`:

```
$ ./manage syncdb
```

dan outputnya kurang lebih sebagai berikut:

```

Creating tables ...
Creating table auth_permission
Creating table auth_group_permissions
Creating table auth_group
Creating table auth_user_user_permissions
Creating table auth_user_groups
Creating table auth_user
Creating table django_content_type
Creating table django_session
Creating table django_site

```

```

You just installed Django's auth system, which means you don't have
any superusers defined.

```

```

Would you like to create one now? (yes/no): yes

```

```

Username (leave blank to use 'banthink'): admin

```

```

E-mail address: admin@example.com

```

```

Password:

```

```

Password (again):

```

```

Superuser created successfully.

```

```

Installing custom SQL ...

```

```

Installing indexes ...

```

```

Installed 0 object(s) from 0 fixture(s)

```

6 Mengaktifkan Modul Admin

Sangatlah mudah mengaktifkan modul Admin pada Django, berikut langkah-langkahnya:

Buka file `settings.py`, cari bagian berikut:

```

INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # Uncomment the next line to enable the admin:
    # 'django.contrib.admin',
    # Uncomment the next line to enable admin documentation:
    # 'django.contrib.admindocs',
)

```

Uncomment baris berikut dengan cara menghilangkan tanda '#'

```
'django.contrib.admin',
```

Setelah itu, edit file `urls.py`:

```

from django.conf.urls import patterns, include, url

# Uncomment the next two lines to enable the admin:
# from django.contrib import admin
# admin.autodiscover()

urlpatterns = patterns('',
    # Examples:
    # url(r'^$', 'mysite.views.home', name='home'),
    # url(r'^mysite/', include('mysite.foo.urls')),

    # Uncomment the admin/doc line below to enable admin documentation:
    # url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

    # Uncomment the next line to enable the admin:
    # url(r'^admin/', include(admin.site.urls)),
)

```

Menjadi seperti berikut:

```

from django.conf.urls import patterns, include, url

# Uncomment the next two lines to enable the admin:
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    # Examples:
    # url(r'^$', 'mysite.views.home', name='home'),
    # url(r'^mysite/', include('mysite.foo.urls')),

    # Uncomment the admin/doc line below to enable admin documentation:
    # url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

    # Uncomment the next line to enable the admin:
    url(r'^admin/', include(admin.site.urls)),
)

```

Ada 3 baris yang harus di *uncomment*:

```
from django.contrib import admin
admin.autodiscover()

url(r'^admin/', include(admin.site.urls)),
```

Jalankan kembali server Anda dengan perintah:

```
(testing)$ ./manage.py runserver
```

Sekarang Anda dapat mencoba mengakses *url admin* dengan mengetikkan di browser:

`http://127.0.0.1/admin`

Jika tidak ada yang salah, seharusnya Anda sekarang dibawa ke halaman Django login, silahkan coba login sesuai dengan username dan password yang sudah Anda tentukan di awal tadi.

7 Creating Your First App

Setelah ini, Anda dapat mulai berlatih dengan mencoba tutorial dari Django Documentation²

Happy Coding :)

²<https://docs.djangoproject.com/en/1.4/intro/tutorial01/>