

---

# Asas Django

*Release 0.1*

**Kamal**

May 22, 2012



# CONTENTS

<b>1</b>	<b>Keperluan</b>	<b>3</b>
<b>2</b>	<b>Instalasi</b>	<b>5</b>
2.1	Cuba Instalasi . . . . .	12
<b>3</b>	<b>Aplikasi Asas</b>	<b>15</b>
3.1	Lancar Aplikasi . . . . .	16
<b>4</b>	<b>Indices and tables</b>	<b>23</b>



## Asas Pengaturcaraan Web Menggunakan Python/Django

### Contents:

Panduan ringkas ini adalah untuk mereka yang berminat untuk mempelajari kaedah pembangunan laman web atau aplikasi web menggunakan *framework* Django. Django adalah *web framework* yang dibangunkan menggunakan bahasa pengaturcaraan Python.



# KENAPA PYTHON/DJANGO ?

Anda mungkin bertanya-tanya kenapa perlu belajar Python berbanding bahasa pengaturcaraan yang lain ? Python adalah antara bahasa pengaturcaraan umum (*general purpose programming language*) yang kian meningkat adopsinya dalam bidang pembangunan perisian. Anda boleh melihat penggunaannya bermula daripada peranti-peranti kecil seperti telefon bimbit sehingga dalam bidang penyelidikan di [NASA](#) dan [lain-lain organisasi penyelidikan](#).

Salah satu kelainan utama Python adalah pada sintaksnya yang sedikit berbeza berbanding bahasa pengaturcaraan yang anda biasa jumpa. Python menggunakan *whitespace* sebagai cara untuk membezakan setiap *block* kod berbanding penggunaan token khas seperti 'begin, end', '{, }' dalam bahasa pengaturcara lain. Contohnya:

```
# Python
if is_logged_in:
    print 'Selamat datang'

// PHP
if ($is_logged_in) {
    print 'Selamat datang';
}

// PHP
if ($is_logged_in) {
print 'Selamat datang';
}

// PHP
if ($is_logged_in) {print 'Selamat datang';}

// Ruby
if is_logged_in
    print 'Selamat datang'
end
```

Ini bermakna indentasi yang betul adalah sesuatu yang wajib dalam Python. Anda mungkin merasakan ia suatu yang melecehkan namun apabila anda perlu membaca kembali kod yang ditulis sama ada oleh anda sendiri atau orang lain, anda akan dapati kod Python lebih senang untuk dibaca dan difahami. Ini sangat penting kerana sedar atau tidak 80% masa dalam pengaturcaraan adalah dalam membaca kod yang telah ditulis berbanding menulisnya. Malah dalam kebanyakan projek yang menggunakan bahasa selain Python, anda tetap diminta untuk meng'indent' kod supaya lebih senang dibaca dan difahami.

Selain itu Python juga mempunyai koleksi *builtin standard libraries* yang secara relatifnya amat komprehensif. *Library* untuk memanipulasi database sqlite, JSON, XML, mencapai *network resources*, GUI toolkit, kriptografi, *data compression*, *multithreading/multiprocess* dan banyak lagi telah disertakan dalam Python.

## 1.1 Django

Django adalah *framework* khusus untuk membangun aplikasi web menggunakan Python. Antara lain, ia di-datangkan dengan *tools* seperti:-

- ORM
- Automatic admin interface
- Template system
- User Authentication
- Session Management
- Form validation
- Internationalization - untuk pembangunan website dalam pelbagai bahasa.
- Cache system
- GIS/Spatial data
- Pagination
- Email sending framework
- Security, Cryptographic signing
- Comments system



# KEPERLUAN

Untuk panduan ini, saya menganggap pembaca akan menggunakan sistem operasi (OS) Windows XP atau yang terkini. Aplikasi yang akan digunakan adalah seperti berikut:-

- [PortablePython](#)
- [Notepad++](#)
- Pelayar Web (Firefox/Google Chrome/Internet Explorer dll)



## INSTALASI

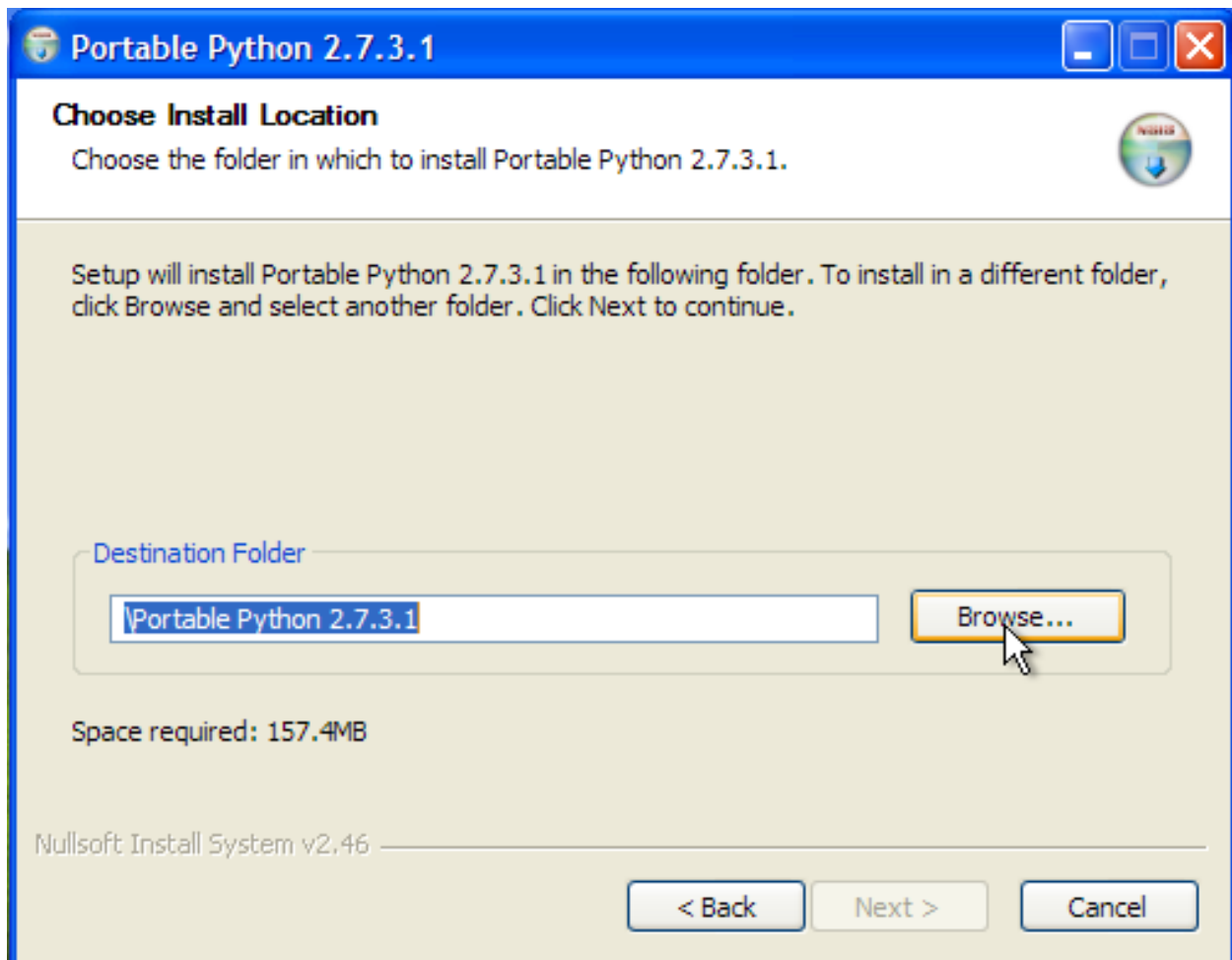
Kita mulakan dengan instalasi PortablePython. PortablePython adalah satu pakej Python beserta dengan *libraries* seperti Django, PIL (Python Imaging Libraries) dll yang mudah untuk di'install' pada komputer anda. URL untuk memuat-turun aplikasi ini adalah:-

<http://portablepython.com/wiki/PortablePython2.7.3.1>

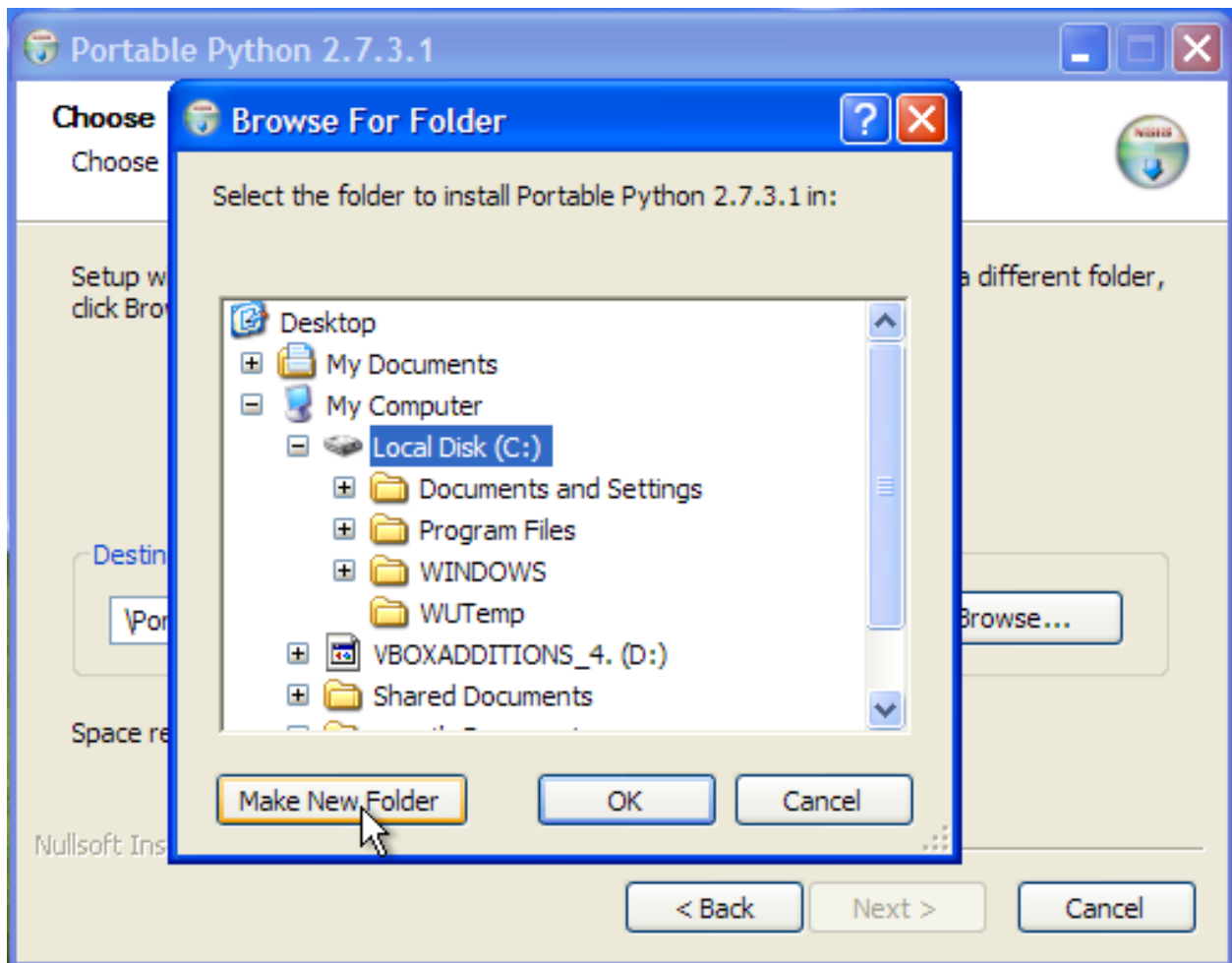
Sila pilih *mirror* di bahagian bawah laman tersebut untuk memulakan proses muat-turun. Apabila selesai muat-turun, klik dua kali pada fail tersebut untuk melancarkan *installer*. Anda akan mendapat skrin seperti berikut:-

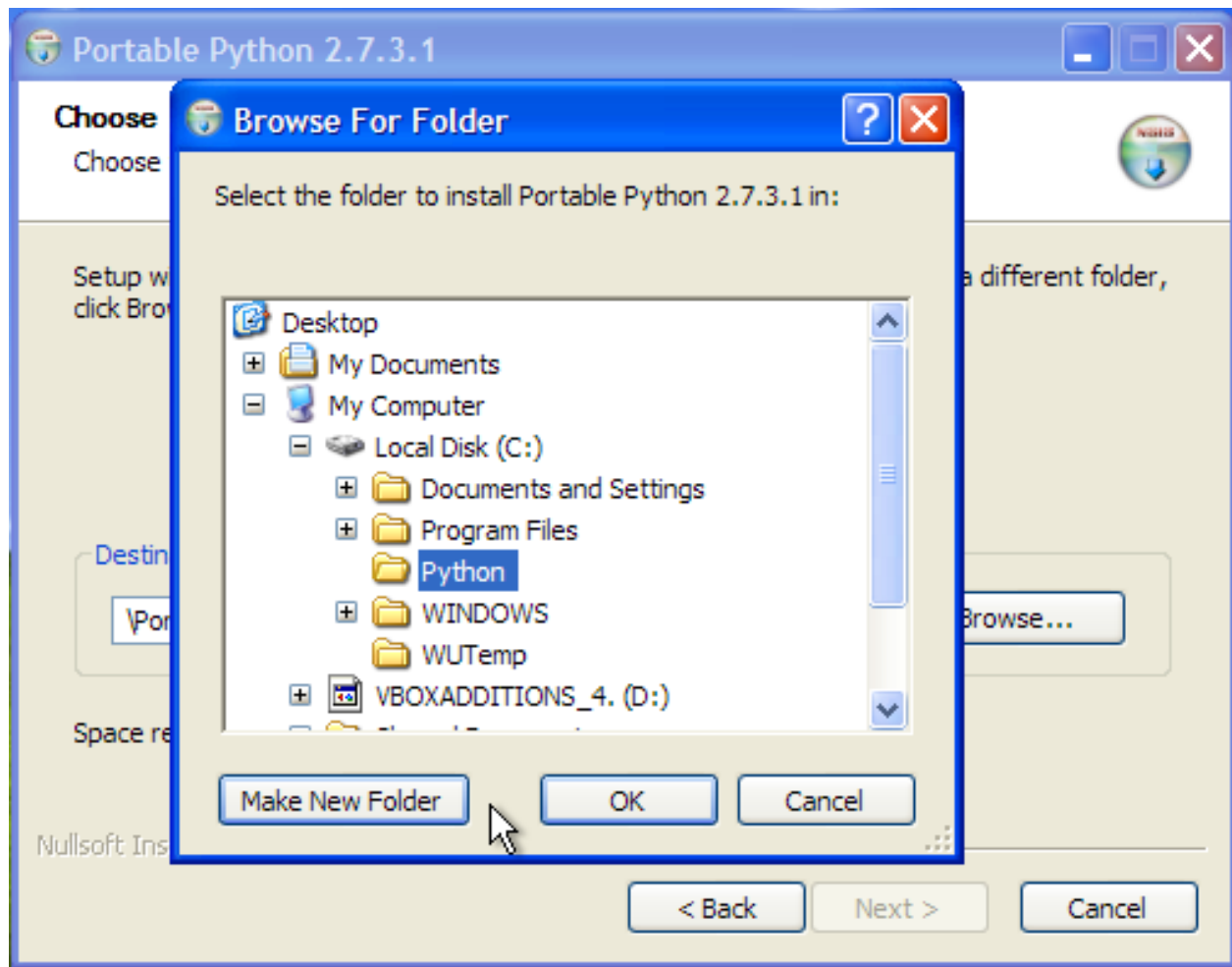


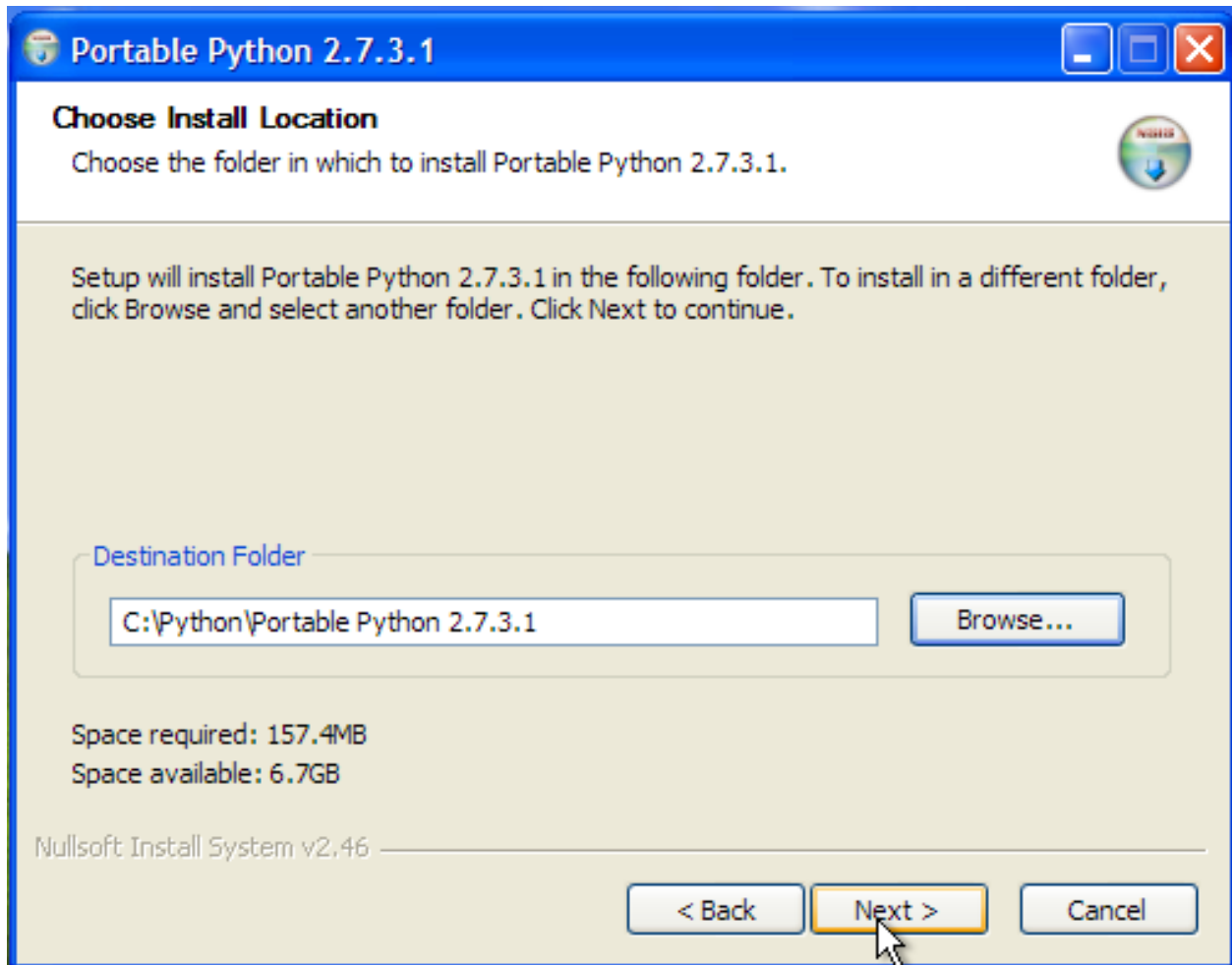
Klik butang *next* dan anda seharusnya mendapat skrin berikut:-



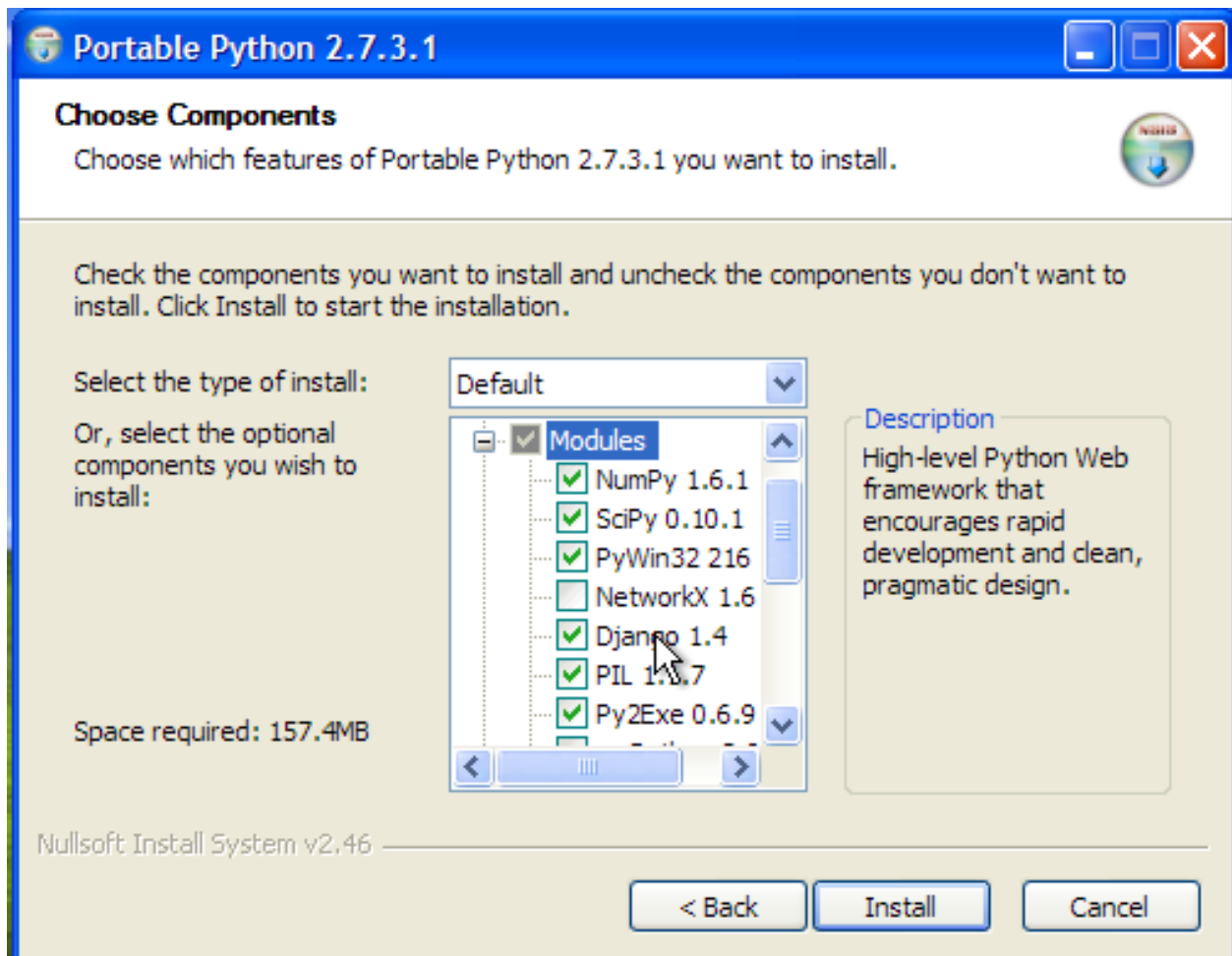
Klik tombol *browse* untuk memilih lokasi *folder* untuk pemasangan. Saya cadangkan anda *install* aplikasi ini di folder *C:\Python*. Berikut cekupan skrin untuk step ini:-





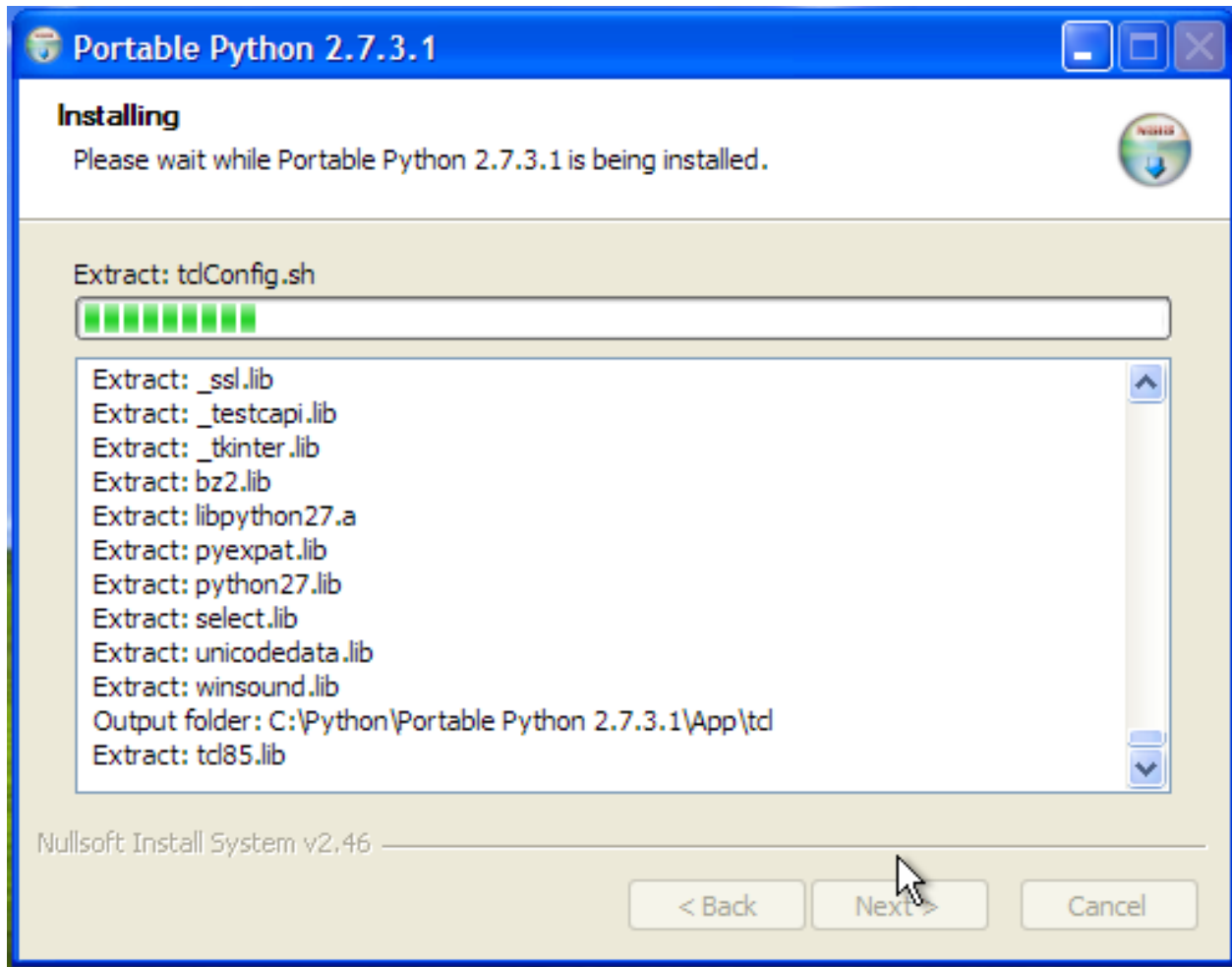


Setelah menetapkan lokasi *folder*, langkah berikutnya adalah untuk memilih *libraries* untuk dimasukkan sekali dalam pemasangan ini. Sila pastikan Django dipilih:-



Langkah terakhir adalah dengan menekan tombol *Install* bagi memulakan proses instalasi. Anda akan melihat skrin berikut dipaparkan:-



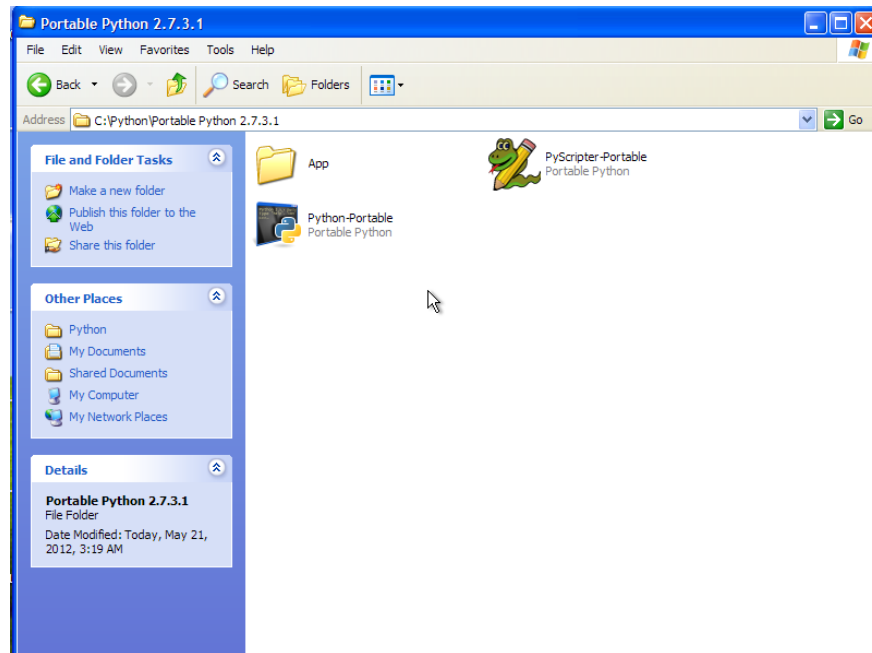


Proses instalasi akhirnya tamat.

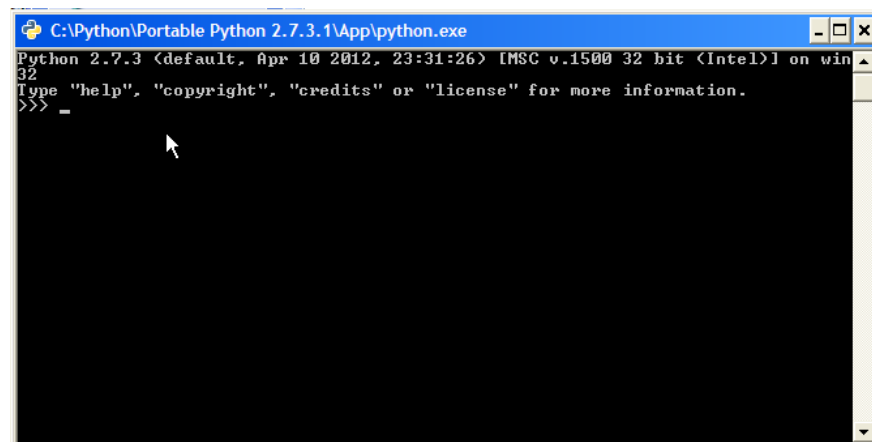


### 3.1 Cuba Instalasi

Sekarang mari kita pastikan proses instalasi berjalan dengan lancar dan kita mendapat pakej `Python` yang diperlukan. Navigasi ke dalam `My Computer` anda ke lokasi berikut - `C:\Python\Portable Python 2.7.3.1`. Pastikan *folder* tersebut mempunyai folder dan fail-fail seperti dalam cekupan skrin di bawah:-



Klik dua kali pada fail Portable-Python untuk melancarkannya dan anda akan mendapat skrin konsol Python.

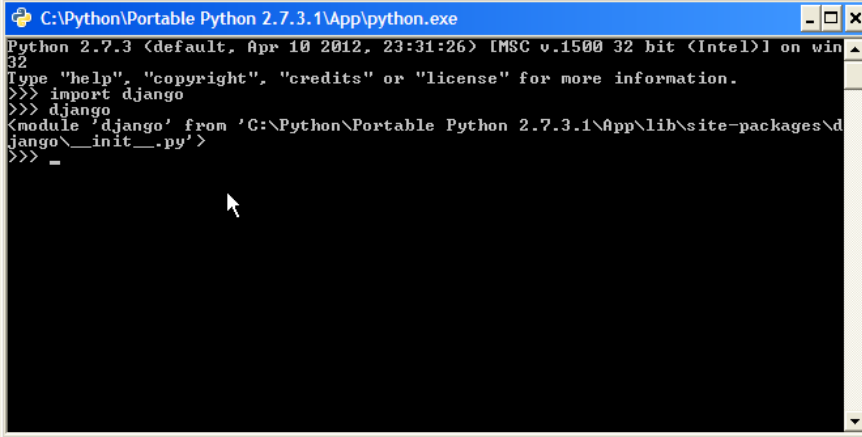


Taip kod berikut dalam konsol berkenaan:

```
>>> import django
>>> django
```

Anda akan mendapat output seperti berikut:

```
<module 'django' from 'C:\Python\Portable Python 2.7.3.1\App\lib\site-packages\django\__init__.pyc'>
```



```
C:\Python\Portable Python 2.7.3.1\App\python.exe
Python 2.7.3 <default, Apr 10 2012, 23:31:26> [MSC v.1500 32 bit <Intel>] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> import django
>>> django
<module 'django' from 'C:\Python\Portable Python 2.7.3.1\App\lib\site-packages\d
jango\__init__.py'>
>>> _
```

Tahniah ! Ini menunjukkan anda telah berjaya meng'*install*' sistem Python ke dalam komputer anda. Anda sekarang sudah bersedia untuk mula mempelajari Django.

# APLIKASI ASAS

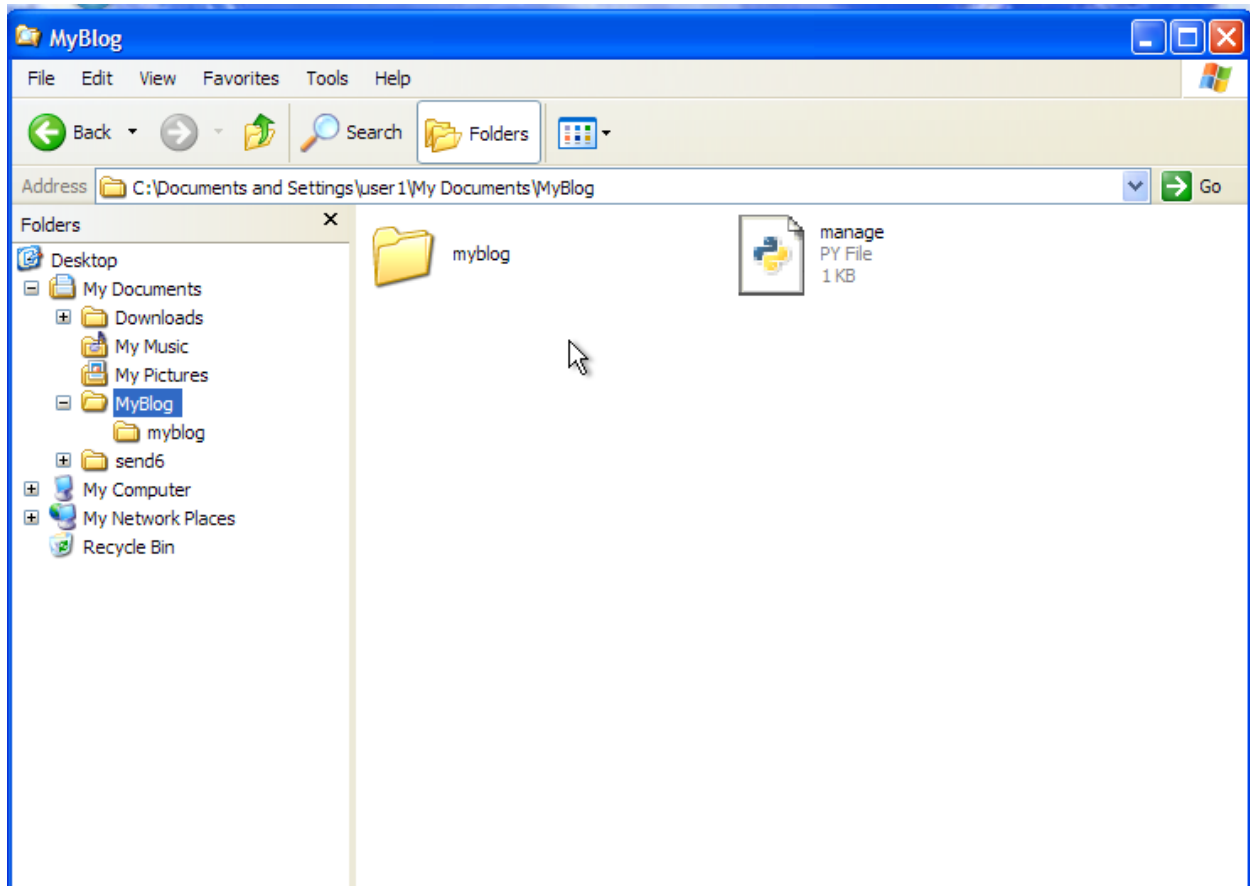
Dalam bahagian ini, kita akan mula membina struktur asas sebuah aplikasi berasaskan Django. Kita akan mencuba melancarkan aplikasi Django yang boleh diakses melalui pelayar web menggunakan alamat - <http://localhost:8000/> dan ia akan memaparkan teks “Selamat Datang !”.

---

**Note:** Pendekatan yang digunakan dalam panduan ini sedikit berlainan daripada yang digunakan oleh tutorial FIXME::rasmi daripada project [Django](#) di mana mereka menggunakan skrip `django-admin.py` untuk *generate* struktur asas aplikasi. Penulis bagaimanapun berpendapat struktur yang dijana oleh `django-admin.py` mengandungi terlalu banyak kod yang boleh mengganggu proses pemahaman mereka yang baru hendak bermula.

---

Kita namakan projek [Django](#) pertama ini sebagai **MyBlog**. Projek ini kita tempatkan dalam folder `My Documents\MyBlog` jadi mulakan dengan membuat direktori tersebut dalam folder `My Documents` anda. Hasil akhir perlu kelihatan seperti berikut:-



Di sini anda dapat lihat saya telah membuat satu lagi folder yang dinamakan `myblog` dan satu fail yang dinamakan `manage.py`.

**Note:** Konfigur *Windows Explorer* anda untuk memaparkan *extension* bagi setiap fail. Dalam Windows XP, ia boleh dilakukan melalui `Tools -> Folder Options -> View -> Advance Settings`.

Mari saya terangkan apakah kegunaan fail-fail dan folder ini secara ringkas:-

- `myblog` - Folder ini merupakan *package* dalam *Python*. Di sinilah kebanyakan daripada kod-kod aplikasi akan ditempatkan. Dalam kod *Python* anda akan merujuk kepada *package* ini melalui *import statement* seperti:  

```
import myblog
```
- `manage.py` - Ini adalah *launcher* kepada aplikasi kita. Dalam *Django* ia dikenali sebagai *management script* kerana selain untuk *execute* aplikasi, ia juga digunakan untuk tugas-tugas seperti *generate database tables*, *copy files* dan lain-lain.

Folder `MyBlog` yang mengandungi fail-fail serta folder ini pula boleh dianggap sebagai folder projek kerana ia akan turut mengandungi fail-fail selain kod sumber seperti dokumentasi, *README*, skrip-skrip pembantu (*helpers script*) dan sebagainya.

## 4.1 Lancar Aplikasi

Saat yang ditunggu-tunggu setelah anda bersusah payah untuk memasang dan konfigur *Python* sehingga berjaya. Kita mulakan dengan menulis kod untuk fail `manage.py`. Saya menggunakan aplikasi *Notepad++* untuk mengedit kod

dalam panduan ini. Anda boleh juga menggunakan aplikasi notepad atau apa-apa sahaja yang serasi.

Taip kod berikut dalam fail `manage.py` anda:

```
from django.conf import settings
from django.core.management import execute_from_command_line

settings.configure(DEBUG=True)
execute_from_command_line()
```

Untuk melancarkan kod yang baru kita tulis ini, kita perlu menggunakan *Python interpreter* yang didatangkan bersama dengan aplikasi PortablePython yang kita *install* dalam bahagian sebelum ini. Anda perlu menggunakan Windows Command Prompt untuk ini. Bukan Command Prompt, jika menggunakan Windows XP ianya adalah melalui butang Start -> All Programs -> Accessories -> Command Prompt. Seterusnya, taip baris arahan berikut:

```
cd "MyDocuments\MyBlog"
"C:\Python\Portable Python 2.7.3.1\app\python.exe" manage.py runserver
```

**Note:** Baris arahan di atas menganggap anda memasang aplikasi PortablePython ke folder `C:\Python\Portable Python 2.7.3.1`. Jika anda memilih lokasi berlainan, sila betulkan *path* berkaitan.

Outputnya harus kelihatan seperti dalam cekupan skrin dibawah:-

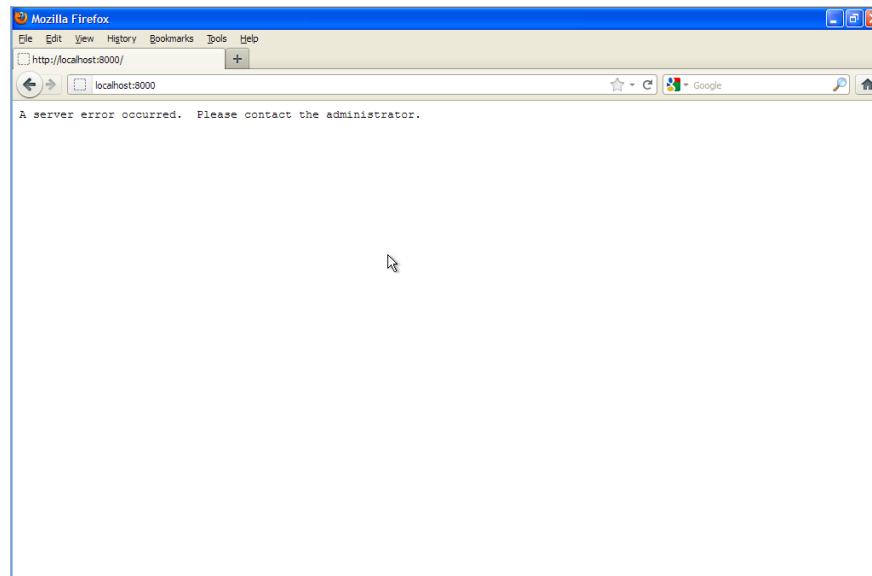
```
C:\Documents and Settings\user1>cd "My Documents\MyBlog"
C:\Documents and Settings\user1\My Documents\MyBlog>dir
Volume in drive C has no label.
Volume Serial Number is CC10-F5AB

Directory of C:\Documents and Settings\user1\My Documents\MyBlog
05/21/2012  07:12 AM    <DIR>          .
05/21/2012  07:12 AM    <DIR>          ..
05/22/2012  04:28 AM             537 manage.py
05/21/2012  04:35 AM    <DIR>          myblog
               1 File(s)              537 bytes
               3 Dir(s)  6,827,462,656 bytes free

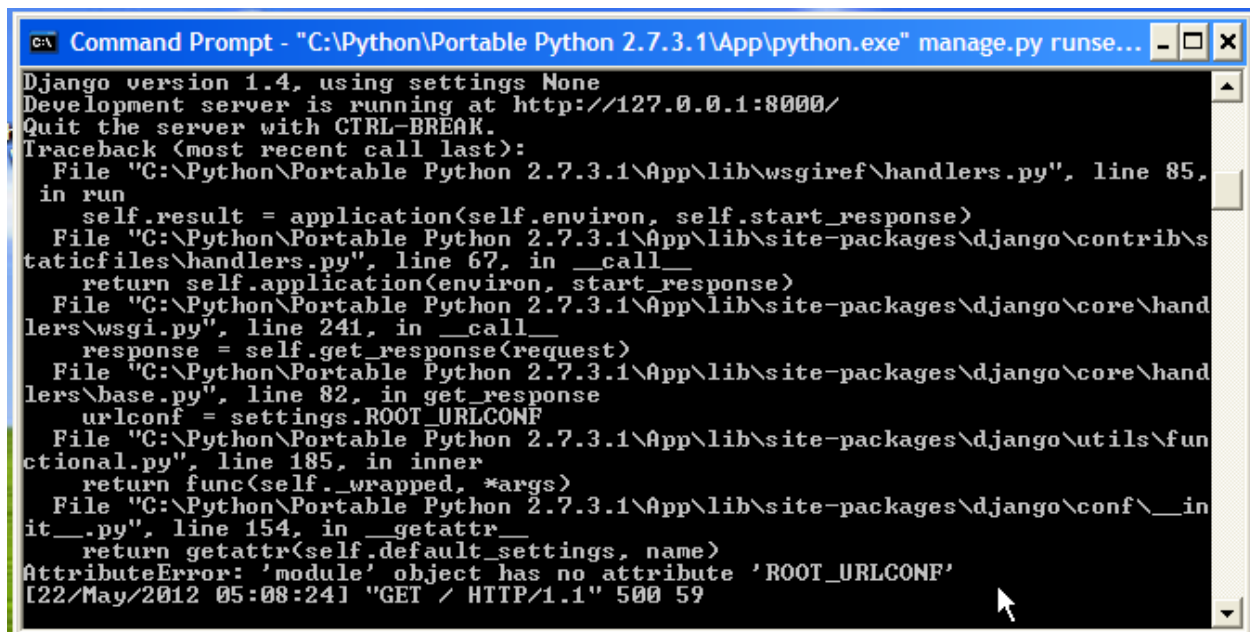
C:\Documents and Settings\user1\My Documents\MyBlog>"C:\Python\Portable Python 2
.7.3.1\app\python.exe" manage.py runserver
Validating models...

0 errors found
Django version 1.4, using settings None
Development server is running at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Sekarang buka aplikasi browser anda (saya menggunakan FireFox di sini) dan cuba akses alamat berikut - <http://localhost:8000>. Browser anda akan memaparkan laman berikut:-



Ada *error* nampaknya. Jangan risau. Kod anda masih belum lengkap sepenuhnya jadi ianya bukanlah memeranjatkan. Mari kita lihat dalam Windows Command Prompt jika ada sebarang output yang boleh membantu kita untuk *debug* masalah yang kita hadapi sekarang.

A screenshot of a Windows Command Prompt window. The title bar reads 'Command Prompt - "C:\Python\Portable Python 2.7.3.1\App\python.exe" manage.py runse...'. The output shows the Django version (1.4), the settings (None), and the server running at http://127.0.0.1:8000/. It then shows a traceback for a server error. The error is an AttributeError: 'module' object has no attribute 'ROOT\_URLCONF'. The traceback points to the django.conf module in the settings file.

```
C:\Python\Portable Python 2.7.3.1\App\python.exe manage.py runse...
Django version 1.4, using settings None
Development server is running at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
Traceback (most recent call last):
  File "C:\Python\Portable Python 2.7.3.1\App\lib\wsgiref\handlers.py", line 85,
    in run
    self.result = application(self.environ, self.start_response)
  File "C:\Python\Portable Python 2.7.3.1\App\lib\site-packages\django\contrib\
    staticfiles\handlers.py", line 67, in __call__
    return self.application(environ, start_response)
  File "C:\Python\Portable Python 2.7.3.1\App\lib\site-packages\django\core\hand
    lers\wsgi.py", line 241, in __call__
    response = self.get_response(request)
  File "C:\Python\Portable Python 2.7.3.1\App\lib\site-packages\django\core\hand
    lers\base.py", line 82, in get_response
    urlconf = settings.ROOT_URLCONF
  File "C:\Python\Portable Python 2.7.3.1\App\lib\site-packages\django\utils\fun
    ctional.py", line 185, in inner
    return func(self._wrapped, *args)
  File "C:\Python\Portable Python 2.7.3.1\App\lib\site-packages\django\conf\__in
    it__.py", line 154, in __getattr__
    return getattr(self.default_settings, name)
AttributeError: 'module' object has no attribute 'ROOT_URLCONF'
[22/May/2012 05:08:24] "GET / HTTP/1.1" 500 59
```

Perhatikan pada bahagian bawah sekali, anda akan mendapat baris yang kelihatan seperti berikut:

```
AttributeError: 'module' object has no attribute 'ROOT_URLCONF'
```

ROOT\_URLCONF adalah salah satu *settings* yang diperlukan oleh Django dan nampaknya kita tidak menyediakan *settings* tersebut. Buka semula fail `manage.py` menggunakan **Notepad++**. Kita perlu membetulkan kod dalam fail ini supaya kelihatan seperti berikut:

```
from django.conf import settings
from django.core.management import execute_from_command_line

settings.configure(DEBUG=True, ROOT_URLCONF='myblog.urls')
```



```
execute_from_command_line()
```

---

**Note:** Kesemua *settings* yang Django perlukan kita akan *pass* sebagai *parameter* kepada *function* `settings.configure()`. Sebenarnya ada 2 cara untuk *pass settings* kepada Django tapi untuk sementara waktu mari kita gunakan cara ini sahaja.

---

*Settings* `ROOT_URLCONF` adalah untuk memberitahu Django akan *path* kepada *module* Python yang akan bertindak sebagai *router* dalam aplikasi kita. *URL Routing* dalam sebuah web aplikasi berfungsi memadankan URL daripada browser kepada *function* atau apa-apa bentuk *handler* dalam aplikasi kita. *Function* atau *handler* ini kemudiannya bertanggungjawab untuk mengeluarkan output yang akan dapat dipaparkan pada browser.

*Module* dalam Python adalah apa-apa fail yang mempunyai *extension* `.py` dan mengandungi kod Python. Untuk itu, mari kita masukkan kod asas bagi *URL routing* bagi aplikasi kita. Kod ini perlu dimasukkan ke fail `myblog\urls.py`:

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

import myblog

urlpatterns = patterns('',
    url(r'^$', myblog.views.index),
)
```

Dalam kod di atas, kita ada merujuk kepada module `myblog.views` jadi untuk melengkapkan kod di atas, kita sediakan juga kod untuk module `myblog.views` ke dalam fail `myblog\views.py`:

```
from django.http import HttpResponse

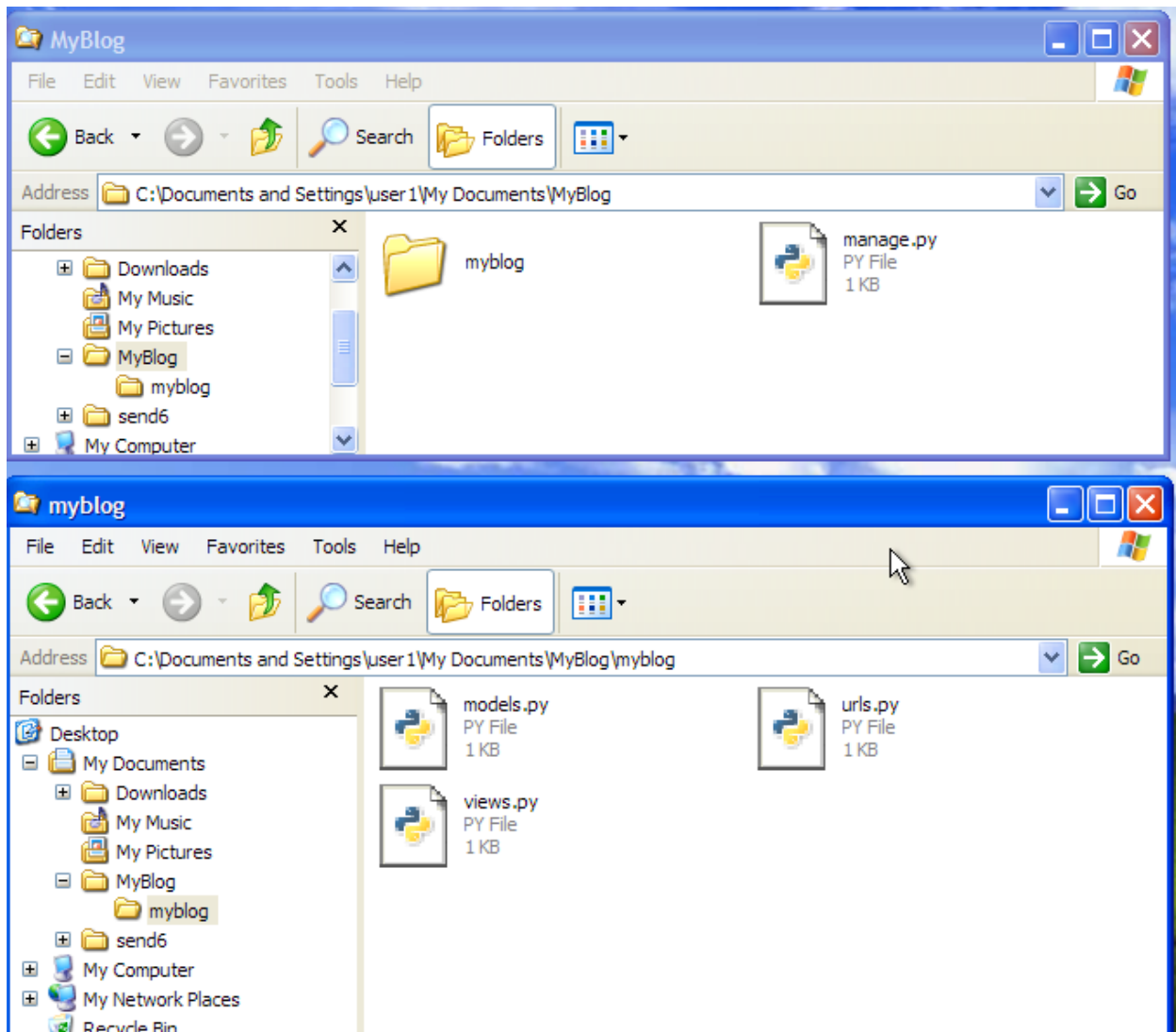
def index(request):
    return HttpResponse('Selamat Datang !')
```

---

**Note:** Dalam Python, untuk merujuk kepada module dalam sesuatu kod, kita akan ganti *separator* untuk folder seperti `\` atau `/` dalam Unix/Linux kepada `.` (dot) dan buang *extension* `.py`. Contohnya untuk module `myblog\views.py` apabila dirujuk melalui kod akan menjadi `myblog.views` sahaja.

---

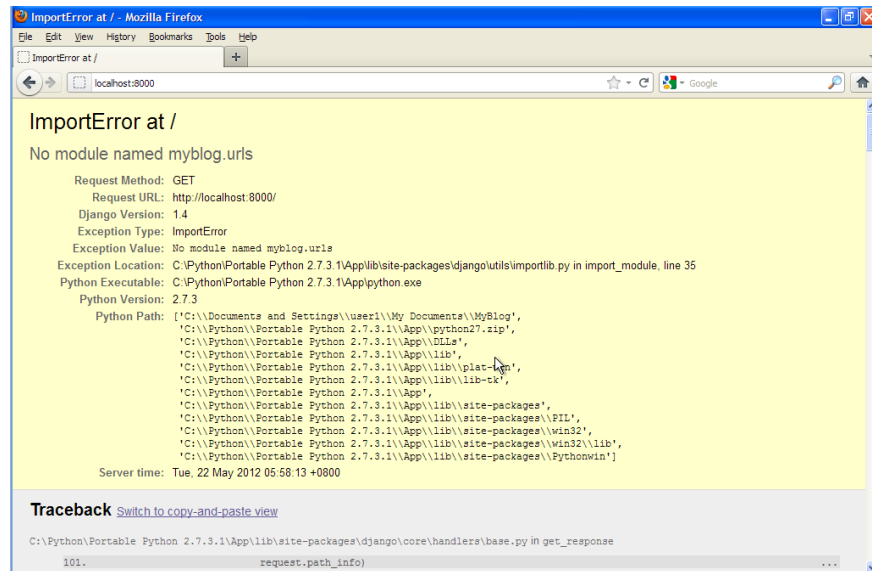
Hasil akhir, folder `myblog` kita akan mempunyai fail seperti berikut:-



Mari kita coba lancarkan aplikasi ini sekali lagi. Seperti sebelum ini, taip arahan berikut dalam Windows Command Prompt:

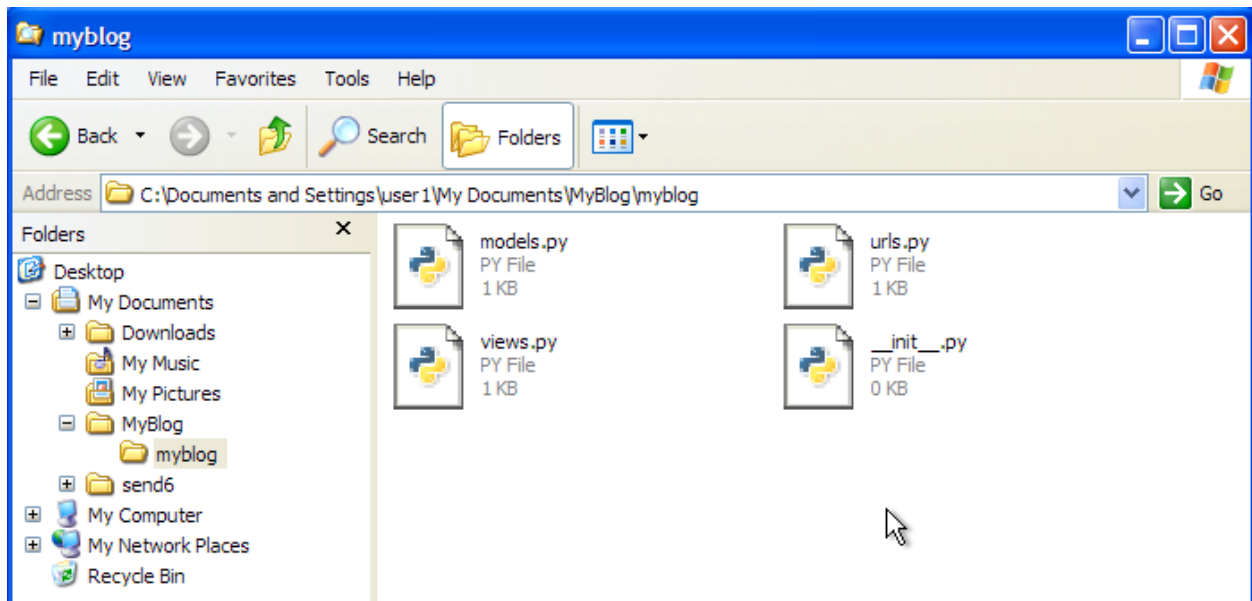
```
"C:\Python\Portable Python 2.7.3.1\app\python.exe" manage.py runserver
```

Cuba akses alamat <http://localhost:8000/> melalui pelayar web anda. Anda akan dapat paparan seperti berikut:-

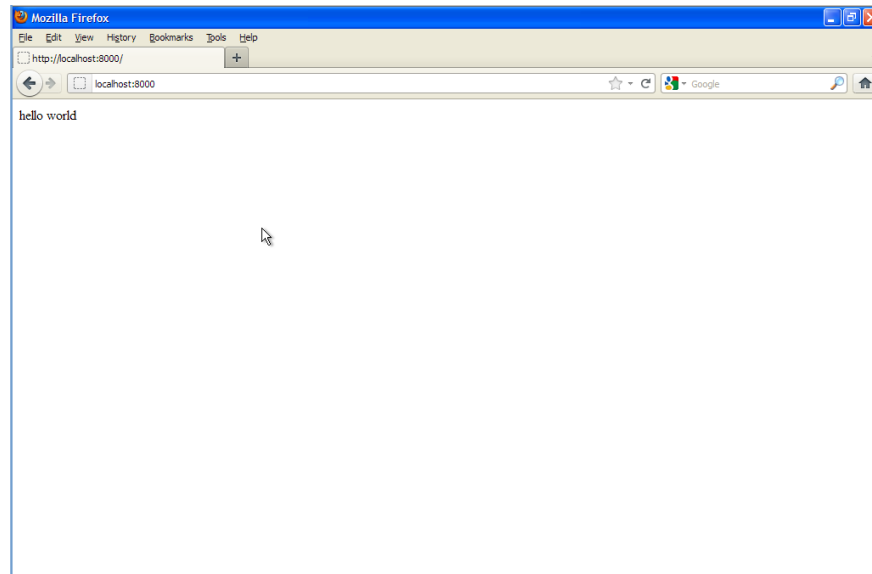


Error lagi ! Saya harap anda bersabar. Memulakan pengaturcaraan dalam satu bahasa dan *framework* yang baru memang memakan masa dan memerlukan sedikit kesabaran. Mari kita cuba debug masalah yang kita hadapi sekarang. Mesej yang diberikan oleh Django adalah ImportError. Ini biasanya berlaku apabila Python gagal untuk menemui module yang cuba dirujuk dalam sesuatu kod Python. Dalam kes ini, module yang gagal ditemui adalah `myblog.urls`. Ini agak pelik kerana kita telah pun menyediakan fail `myblog\urls.py` sebelum ini.

Untuk tidak membuang masa, saya nyatakan saja apa masalah sebenarnya. Di dalam folder `myblog`, kita perlu sediakan satu lagi fail yang perlu dinamakan sebagai `__init__.py`. Fail ini adalah untuk memberitahu Python bahawa folder tersebut adalah merupakan satu *package*. Anda tidak perlu memasukkan apa-apa kod ke dalam fail ini. Untuk kebanyakan masa, fail ini adalah kosong. Folder `myblog` perlu kelihatan seperti berikut sekarang:-



Reload browser anda dan akhirnya anda akan dapat output yang diharapkan seperti berikut:-



# INDICES AND TABLES

- *genindex*
- *modindex*
- *search*