

*Skripsi Geofisika*

# **RANCANG BANGUN WEB-GIS BERBASIS GEODJANGO-PYTHON**



**DISUSUN OLEH:**

**AR ADITYA HASANUDDIN**

**H221 16 514**

**DEPARTEMEN GEOFISIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2020**



# **RANCANG BANGUN WEB-GIS BERBASIS**

## **GEODJANGO-PYTHON**

Skripsi ini untuk melengkapi tugas-tugas dan memenuhi syarat untuk mencapai  
gelar sarjana



**AR ADITYA HASANUDDIN**

**H221 16 514**

**DEPARTEMEN GEOFISIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2020**



# **RANCANG BANGUN WEB-GIS BERBASIS GEODJANGO-PYTHON**

**OLEH:**

**AR ADITYA HASANUDDIN**

**H221 16 514**

**SKRIPSI**

**Untuk memenuhi salah satu syarat ujian**

**Guna memperoleh gelar Sarjana Sains**

**Program Pendidikan Sarjana, Departemen Geofisika ini**

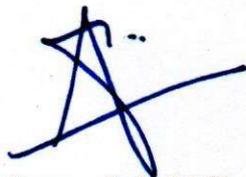
**Telah disetujui oleh Tim Pembimbing pada tanggal**

**Seperti tertera di bawah ini**

**Makassar, 25 Agustus 2020**

**Disetujui Oleh:**

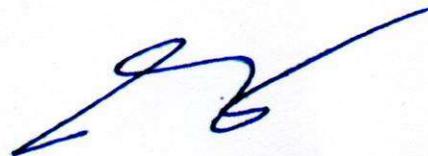
**Pembimbing Utama**



**Dr. Samsu Arif, M.Si**

**196305181991031011**

**Pembimbing Pertama**



**Dr. Muh. Alimuddin Hamzah, M.Eng**

**196709291993031003**



## SURAT PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : AR Aditya Hasanuddin

NIM : H221 16 514

Departemen : Geofisika

Judul TA : Rancang Bangun *web-gis* Berbasis GeoDjango-Python

Menyatakan bahwa skripsi ini benar-benar hasil karya saya sendiri dan belum pernah diajukan untuk mendapatkan gelar sarjana di Universitas Hasanuddin atau Lembaga Penelitian lain kecuali kutipan dengan mengikuti tata penulisan karya ilmiah yang sudah lazim digunakan, karya tulis ini merupakan murni gagasan dan penelitian saya sendiri, kecuali arahan Tim Pembimbing dan masukan Tim Penguji.

Makassar, 25 Agustus 2020

Saya Membuat Pernyataan



**AR Aditya Hasanuddin**



## ABSTRAK

### RANCANG BANGUN WEB-GIS BERBASIS GEODJANGO-PYTHON

Web-GIS merupakan sistem informasi geografis yang terdiri dari gabungan antara desain grafis pemetaan, peta digital dengan analisis geografis, pemrograman komputer, dan sebuah database yang saling terhubung menjadi satu bagian web pemetaan. Penelitian ini bertujuan untuk membuat sistem informasi geografis berbasis web kemudian di implementasikan menggunakan *framework* GeoDjango-Python. Penelitian ini menggunakan *framework* GeoDjango sebagai kerangka dasar dari pembuatan *web-GIS* dan menggunakan PostgreSQL yang memiliki ekstensi PostGIS sebagai penyimpanan data geospasial. Hasil penelitian ini berupa sebuah aplikasi sistem informasi geografis berbasis web yang dapat menampilkan data peta ke dalam sebuah website geografi dengan menggunakan *web framework* GeoDjango.

**Kata Kunci:** Python, *web-GIS*, GeoDjango, PostGIS.



## ABSTRACT

### GEODJANGO-PYTHON BASED WEB-GIS DESIGN

Web-GIS is a geographic information system consisting of a combination of graphical mapping design, digital maps with geographic analysis, computer programming, and a database interconnected into one Web Part mapping. The research aims to create a Web-based geographic information system then implemented using the GeoDjango-Python framework. The research uses the GeoDjango framework as the basic framework of web-GIS creation and uses PostgreSQL which has PostGIS extensions as a geospatial data store. The results of this study are a web-based geographic information system application that can display map data into a geographic website using geodjango web framework.

**Keywords:** Python, *web*-GIS, GeoDjango, PostGIS.



## KATA PENGANTAR

**“Assalamualaikum Warahmatullahi Wabarakatuh”**

Segala puji dan syukur penulis panjatkan kepada Allah SWT atas berkah, rahmat, dan hidayah-Nya yang senantiasa dilimpahkan kepada penulis sehingga dapat menyelesaikan penyusunan skripsi yang berjudul **“Rancang Bangun *web*-GIS Berbasis GeoDjango-Python”** sebagai syarat untuk menyelesaikan Program Sarjana (S1) pada Program Sarjana Fakultas Matematika dan Ilmu Pengetahuan Alam Jurusan Geofisika Universitas Hasanuddin. Shalawat beserta salam semoga senantiasa terlimpah kepada Rasulullah SAW. dalam penyusunan skripsi ini tidak terlepas dari hambatan serta rintangan yang penulis hadapi namun pada akhirnya dapat melaluinya berkat Ridho Allah SWT dan bimbingan dari berbagai pihak. Sebelumnya izinkan penulis untuk ucapkan terima kasih sedalam-dalamnya kepada orang tua tercinta **Ayahanda Drs. Hasanuddin. M.Si**, dan **ibunda A. Maemunah AM (almh)**, kakak tersayang **AR Fuad Hasanuddin**, **AR Basri Hasanuddin**, dan **AR Pratiwi Hasanuddin**, beserta adik tersayang **AR Mubarak Hasanuddin**, **AR Maruf Hasanuddin**, dan **AR Hidayat Hasanuddin**, dan juga kepada semua keluarga besar yang tak henti-hentinya mendoakan dan memberikan dukungan kepada penulis.

Penulis menyadari bahwa penyusunan skripsi ini tidak dapat terselesaikan dengan baik tanpa dukungan dan bantuan dari berbagai pihak baik moril maupun material.

Sehingga itu dalam kesempatan ini pula penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada yang terhormat:



1. **Dr. Samsu Arif, M.Si** selaku Pembimbing Utama dan **Dr. Muh. Alimuddin Hamzah, M.Eng** selaku Pembimbing Pertama yang telah banyak membimbing, meluangkan waktu, memberikan petunjuk serta saran bagi penulis dalam menyusun skripsi.
2. **Dr. Paharuddin, M.Si** dan **Dr. Erfan, M.Si** selaku tim penguji yang telah memberikan kritik, saran, dan masukan yang bersifat membangun kepada penulisan skripsi.
3. **Dr. Eng. Amiruddin, S.Si., M.Si** selaku Dekan Fakultas Matematika dan ilmu Pengetahuan Alam, Universitas Hasanuddin.
4. **Dr. Muh. Alimuddin Hamzah, M.Eng** selaku Ketua Departemen Geofisika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin.
5. **Drs. Hasanuddin, M.Si** selaku Penasehat Akademik yang telah memberikan masukan dan arahan terkait akademik kepada penulis.
6. Seluruh dosen Departemen Geofisika, staf FMIPA UNHAS, staf Departemen Geofisika, staf laboratorium, staf perpustakaan FMIPA UNHAS, dan staf perpustakaan umum atas semua bantuan dan ilmu yang telah diajarkan, pelayanan dan bantuan yang telah diberikan kepada penulis.
7. Seluruh guru-guru SMAN 2 Bulukumba, MTsN 410 Tanete, dan SDN 87 Buttakeke atas ilmu dan pendidikan yang diberikan kepada penulis.
8. Teruntuk **Islamiah Nursalim** teman seperjuangan dalam mengerjakan gas akhir, yang telah menjadi teman bertukar pikiran dan sama-sama emotivasi dalam mengerjakan tugas akhir.



9. Teruntuk **Ayu Agustina, S.Tr.Keb.** yang menjadi motivasi yang selalu membantu dan menjadi penyemangat dalam mengerjakan skripsi ini.
10. Saudara tak sedarah **Iksan Milham, Agung Hasan, S.Si, Aushaf Fauzan DTS, De Leonard Pasteur S, Indra Kurnia Rasa, Muh. Fazrul Rahman, Muh. Azhari Ramlan, Ayyub Alqadri, Muh. Nur Alamsyah Rahman, Muhammad Maulana, Olandiani Pasa'bi, Nur Arief, Nur Aryadint, Nuraisyiah Pertiwi Kamsir, Faradhiba, Mustahira, Sri Wahyuni, Riana Trisartika, Sinar Indriani, S.Si, Nurzakia Tri Utari, S.Si. Addiati, Retno Wulandari, Mutmainnah, Erwinda, Amalia Nur Rachma, Amalia Widyanis, Reski Indrawati Usman, Marhaeni, S.Si.** dan **Maghfira Sulva Salsabila, S.Si.** yang selalu setia membantu penulis dalam segala hal dan menjadi tempat mencurahkan keluhan, bertukar pikiran, berbagi ilmu, suka, dan duka selama dibangku kuliah.
11. Teman-teman seperjuangan **Geofisika 2016 (16NEOUS), HIMAFI FMIPA UNHAS 2016, dan KM FMIPA UNHAS 2016** yang telah memberikan banyak kesan serta membantu dalam mengukir kisah selama kuliah baik di dalam kelas maupun kuliah di lapangan.
12. Sahabat sekaligus keluarga **KKN Reguler Desa Baruga Riattang** yang selalu memberikan semangat dan bantuan selama menjalani KKN.
13. Serta kepada semua pihak yang telah membantu penulis namun tidak sempat penulis sebutkan dalam skripsi ini, sehingga skripsi ini dapat terselesaikan dengan baik, semoga Allah SWT membalas segala perbuatan baik saudara(i) dan menjadi amal ibadah disisi-Nya.



## DAFTAR ISI

HALAMAN SAMPUL .....	i
HALAMAN PENUNJUK SKRIPSI .....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR PERNYATAAN .....	iv
ABSTRAK.....	v
KATA PENGANTAR .....	vii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN .....	1
I.1 Latar Belakang .....	1
I.2 Ruang Lingkup .....	2
I.3 Rumusan Masalah.....	2
I.4 Tujuan.....	2
BAB II TINJAUAN PUSTAKA.....	3
II.1 Internet .....	3
II.2 <i>Web-based GIS</i> .....	3
II.3 Python .....	5
II.3.1 Definisi Phyton.....	5
Fitur Phyton.....	7
ngo.....	9



II.5 GeoDjango .....	12
II.6 MVC Framework.....	13
II.7 Paket <i>Contrib</i> .....	16
II.8 Spasial Database.....	17
II.9 Lingkungan pengembangan proyek Django .....	19
II.9.1 Command Line .....	20
II.9.2 Virtual Environment .....	20
II.9.3 Python Installation Packages .....	22
II.9.4 Proyek Django .....	23
II.10 Geospasial libraries.....	26
II.10.1 GEOS .....	26
II.10.2 PROJ.4.....	26
II.10.3 GDAL.....	26
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>27</b>
III.1 Lokasi Penelitian .....	27
III.2 Alat dan Bahan Penelitian .....	27
III.3 Prosedur Penelitian.....	27
III.3.1 Tahap Persiapan .....	27
III.3.2 Perancangan web-GIS .....	28
III.3.3 Aplikasi Web Geodjango dengan Data di lapangan.....	29



III.4 Bagan alir penelitian.....	30
<b>BAB IV HASIL DAN PEMBAHASAN.....</b>	<b>31</b>
IV.1 Persiapan Instalasi GeoDjango.....	31
IV.2 Instalasi GeoDjango.....	32
IV.3 Pemasangan Spasial Database.....	33
IV.4 Pembuatan Proyek GeoDjango .....	35
IV.5 Pemasangan Modul Pendukung .....	38
IV.6 Pengaturan Settings.py .....	39
IV.7 Pengaturan Models.py.....	40
IV.8 Pembuatan Halaman Utama .....	41
IV.9 Pembuatan Halaman User .....	42
IV.10 Pembuatan Halaman Data List .....	43
IV.11 Pembuatan Halaman Upload, Update Atribut, dan Delete .....	44
IV.12 Data Peta Penelitian .....	46
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>47</b>
V.1 Kesimpulan .....	47
V.2 Saran .....	47
<b>DAFTAR PUSTAKA .....</b>	<b>48</b>
<b>LAMPIRAN.....</b>	<b>50</b>



## DAFTAR GAMBAR

Gambar 2.1 Komponen GIS.....	4
Gambar 2.2 Arsitektur <i>web-based GIS</i> .....	5
Gambar 2.3 Tampilan IDLE Python 3.8.1 .....	6
Gambar 2.4 Pola <i>Model-View-Controller (MVC)</i> .....	14
Gambar 2.5 Alur kerja Django untuk URL, <i>template</i> , dan aplikasi.....	16
Gambar 2.6 Alur kerja <i>PostGIS</i> .....	18
Gambar 2.7 Pemasangan Django menggunakan pip.....	23
Gambar 2.8 Pemasangan Django menggunakan file yang berformat tar.gz.....	24
Gambar 2.9 Struktur proyek Django.....	25
Gambar 4.1 (a) <i>Tampilan utama dari software pgadmin 4 yang merupakan tampilan utama dari PostgreSQL, (b) Ekstensi PostGIS yang telah ditambahkan pada saat instalisasi PostgreSQL</i> .....	33
Gambar 4.2 <i>Halaman utama web-GIS</i> .....	41
Gambar 4.3 (a) <i>Halaman Register, (b) Halaman Login, (c) Halaman Logout</i> ....	42
Gambar 4.4 <i>Halaman Data List</i> .....	43
Gambar 4.5 (a) <i>Halaman Upload peta, (b) Halaman Update Atribut Peta, (c) Halaman delete peta</i> .....	45
Gambar 4.6 <i>Data peta kabupaten Gowa</i> .....	46



## DAFTAR LAMPIRAN

Lampiran 1. Peta Administrasi Kabupaten Gowa.....	50
Lampiran 2. Peta Jalan Kabupaten Gowa .....	51
Lampiran 3. Peta Bangunan Pendidikan Kabupaten Gowa .....	52
Lampiran 4. Data Atribut Peta Administrasi Kabupaten Gowa.....	53
Lampiran 5. Data Atribut Peta Jalan Kabupaten Gowa .....	54
Lampiran 6. Data Atribut Peta Bangunan Pendidikan Kabupaten Gowa .....	55
Lampiran 7. <i>Script Home/Settings.py</i> .....	60
Lampiran 8. <i>Script Home/Urls.py</i> .....	63
Lampiran 9. <i>Script Maps/Models.py</i> .....	64
Lampiran 10. <i>Script Maps/Admin.py</i> .....	67
Lampiran 11. <i>Script Maps/Urls.py</i> .....	68
Lampiran 12. <i>Script Maps/Views.py</i> .....	69
Lampiran 13. <i>Script Users/Forms.py</i> .....	70
Lampiran 14. <i>Script Users/Views.py</i> .....	74
Lampiran 15. <i>Script Template/Home/Index.html</i> .....	78
Lampiran 16. <i>Script Templates/Snippets/Navbar.html</i> .....	83
Lampiran 17. <i>Script Templates/Maps/Administrasi_list.html</i> .....	85



Lampiran 18. <i>Script Templates/Maps/Edubuild_list.html</i> .....	88
Lampiran 19. <i>Script Templates/Maps/Street_list.html</i> .....	91
Lampiran 20. <i>Script Templates/Upload/Upload_administrasi.html</i> .....	94
Lampiran 21. <i>Script Templates/Upload/Upload_edubuild.html</i> .....	98
Lampiran 22. <i>Script Templates/Upload/Upload_street.html</i> .....	102
Lampiran 23. <i>Script Templates/Maps/ Administrasi_update_form.html</i> .....	106
Lampiran 24. <i>Script Templates/Maps/ Edubuild_update_form.html</i> .....	109
Lampiran 25. <i>Script Templates/Maps/ Street_update_form.html</i> .....	112
Lampiran 26. <i>Script Templates/Maps/ Administrasi_delete.html</i> .....	115
Lampiran 27. <i>Script Templates/Maps/ Edubuild_delete.html</i> .....	116
Lampiran 28. <i>Script Templates/Maps/ Street_delete.html</i> .....	117
Lampiran 29. <i>Script Templates/Snippets/Openlayers.html</i> .....	118
Lampiran 30. <i>Script Templates/Snippets/Osm.html</i> .....	119



# BAB I

## PENDAHULUAN

### I.1 Latar Belakang

Perkembangan teknologi merupakan hal yang sangat penting pada era modern ini, sehingga masyarakat perlu mengetahui dan mengerti tentang pentingnya teknologi. Teknologi dapat membantu setiap kegiatan manusia baik dunia akademik, pembangunan, dan lain sebagainya dengan mudah. Untuk meningkatkan kualitas suatu layanan data dan informasi maka diperlukan sebuah aplikasi *website* (Andoyo & Sujarwadi, 2017). Dalam pembuatan sistem informasi geografis berbasis *website*, diperlukan sebuah desain yang dimodelkan menggunakan bahasa pemrograman Python (Solekhan & Supriyo, 2016).

Python merupakan salah satu bahasa pemrograman yang digunakan untuk membangun aplikasi, baik itu berbasis *desktop*, *web* ataupun berbasis *mobile* (Anwar, 2018). Django merupakan sebuah kerangka aplikasi *web* yang dibangun dengan bahasa pemrograman Python. Adanya kerangka ini pengguna dapat membuat *web* dengan lebih cepat, dibandingkan menulis kode dari nol. Django menyertakan sebuah modul *contrib* yang dapat mengubahnya menjadi kerangka *web* geografis yaitu GeoDjango. Geodjang o membuat aplikasi *web* geografis menjadi sesederhana mungkin, seperti layanan berbasis lokasi. Oleh karena itu, penelitian ini berfokus pada perancangan *website* sistem informasi geografis

Geodjango-Python. Dimana dapat menampilkan data spasial berbentuk ng mempunyai ketelitian posisi suatu objek yang cukup baik.



## **I.2 Ruang Lingkup**

Penelitian ini dibatasi dengan pembuatan sistem informasi geografis berbasis *web* menggunakan perluasan *web framework* dari Django yaitu Geodjango.

## **I.3 Rumusan Masalah**

Adapun rumusan masalah dari penelitian ini, yaitu:

1. Bagaimana tahapan pembuatan sistem informasi geografis berbasis *web*?
2. Bagaimana mengimplementasikan sistem informasi geografis berbasis *web* dengan menggunakan *framework* GeoDjango-Python?

## **I.4 Tujuan**

Adapun tujuan dalam penelitian ini, yaitu:

1. Membuat sistem informasi geografis berbasis *web* kemudian di implementasikan menggunakan *framework* GeoDjango-Python.



## BAB II

### TINJAUAN PUSTAKA

#### II.1 Internet

Internet merupakan sebuah jaringan yang menghubungkan komputer satu sama yang lainnya menggunakan *Transmission Control Protocol / Internet Protocol* (TCP/IP) sebagai protocol pertukaran sehingga dapat berkomunikasi, berinteraksi, dan saling bertukar informasi meski dalam jarak yang jauh (Sukaridhoto, 2014).

Alamat IP merupakan representasi dari 32bit bilangan biner yang dituliskan dengan bilangan desimal dengan dibagi menjadi 4 bagian dan dipisahkan menggunakan titik. Contoh bilangan biner dari IP *address* 128.2.7.9, yaitu 10000000 00000010 00000111 00001001, berikut alamat IP yang perlu diperhatikan (Sukaridhoto, 2014):

1. Alamat dengan semua bit = 0, digunakan untuk alamat jaringan (*network address*). Contoh 192.168.1.0, yang dimana bilangan binernya adalah 11000000 10101000 00000001 00000000
2. Alamat dengan semua bit = 1, digunakan untuk alamat *broadcast* (*broadcast address*). Contoh 192.168.1.255, yang dimana bilangan binernya adalah 11000000 10101000 00000001 11111111
3. Alamat *loopback*, alamat dengan IP 127.0.0.0 digunakan sebagai alamat *loopback* dari sistem local

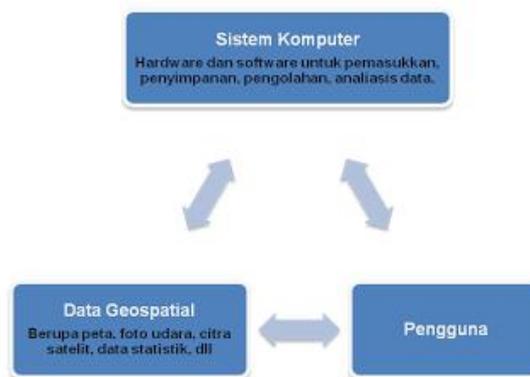


#### *-based GIS*

*Geographical Information System*) adalah sistem berbasis komputer yang n untuk memasukkan, menyimpan, mengelola, menganalisis dan

mengaktifkan kembali data yang mempunyai referensi keruangan untuk berbagai tujuan yang berkaitan dengan pemetaan dan perencanaan. Dalam suatu sistem informasi geografis, terdapat beberapa komponen utama yang saling berintegrasi dan saling terkait, yaitu (Suryo Saputro, 2013):

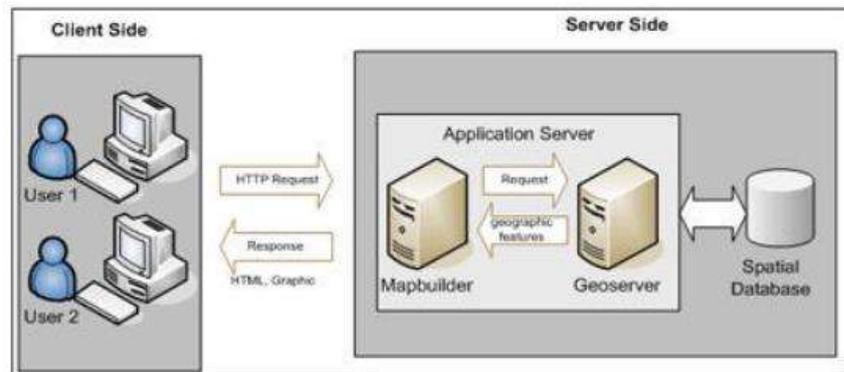
1. Sistem komputer (*hardware* dan *software*)
2. Data geospasial
3. User atau pengguna.



**Gambar 2.1** Komponen GIS (Sumber: Suryo Saputro, 2013).

GIS (*Geographical Information System*) sendiri telah dikembangkan ke arah aplikasi berbasis *web*, atau dikenal dengan *web-GIS*. Tujuannya untuk memudahkan pengguna dalam mendapatkan informasi geografis terkini pada suatu kota atau wilayah tertentu, dan tidak terbatas pada lokasi dari pengguna tersebut.





**Gambar 1.2** Arsitektur *web-based* GIS (Sumber: Suryo Saputro, 2013).

*Web-based GIS* merupakan Sistem Informasi Geografis yang terdiri dari beberapa komponen berbasis *web* yang saling terkait. *Web-based GIS* merupakan gabungan antara design grafis pemetaan, peta digital dengan analisis geografis, pemrograman komputer, dan sebuah database yang saling terhubung menjadi satu bagian web desain dan web pemetaan (Qolis & Fariza, 2010). Secara umum Sistem Informasi Geografis dikembangkan berdasarkan prinsip input, manajemen data, manajemen analisis, dan representasi data. Untuk dapat melakukan komunikasi dengan komponen yang berbeda-beda dilingkungan web maka dibutuhkan sebuah web server (Ramadhani *et al.*, 2016).

## II.3 Python

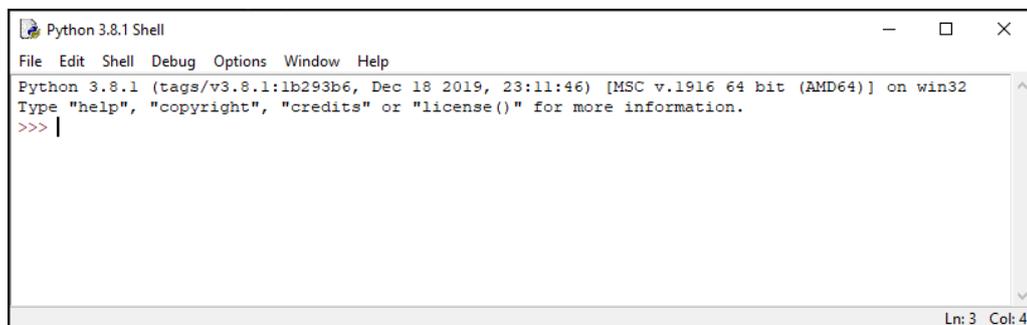
### II.3.1 Definisi Python

Python merupakan bahasa pemrograman yang bersifat *Open Source* sehingga dapat dikembangkan secara gratis tanpa menggunakan lisensi dan dapat dikembangkan sesuai kebutuhan pengguna. Penulisan sintaks pada bahasa pemrograman Python fleksibel sehingga mudah dipelajari. Selain itu, struktur data pada bahasa pemrograman Python sangat efisien, pemrograman yang berorientasi pada objek



lebih sederhana tetapi efektif, dapat bekerja pada *multiplatform*, dan dapat digabungkan dengan bahasa pemrograman lain sehingga menghasilkan aplikasi.

Python merupakan bahasa pemrograman yang di eksekusi dengan cara interpreter. Terdapat dua cara menggunakan interpreter, yaitu dengan mode baris perintah dan modul *script*. Pada mode baris, perintah diketikkan pada *shell* atau *command line* dan di eksekusi secara langsung oleh Python. Cara lain adalah dengan menyimpan perintah-perintah Python dalam satu file, yang disebut selanjutnya sebagai *script*. Kita dapat mengetikkan perintah-perintah Python dengan menggunakan *text editor* seperti *notepad*. Lalu menyimpannya dengan format “.py”. kemudian di eksekusi dengan Python. Pada gambar berikut ditunjukkan tampilan *Integrated Development Environment (IDLE)*. *Prompt >>>* menyatakan interpreter Python siap menerima perintah dari pengguna (Kurniawan *et al.*, 2011).



**Gambar 2.3** Tampilan IDLE Python 3.8.1

Python adalah bahasa pemrograman yang mudah untuk dipelajari dan merupakan bahasa pemrograman yang kuat. Bahasa pemrograman ini memiliki struktur data

yang efisien dan memiliki pendekatan yang sederhana namun efektif. Bahasa pemrograman ini memiliki struktur data yang berorientasi objek, memiliki sintaks yang rapi dan dinamis



sehingga menjadikannya bahasa yang ideal untuk *scripting* dan memiliki pengembangan aplikasi yang cepat dibandingkan kebanyakan *platform* (Swaroop, 2013).

### II.3.2 Fitur Phyton

Menurut Swaroop tahun 2003, Phyton memiliki fitur-fitur yakni:

#### 1. Simple

Phyton merupakan bahasa pemrograman yang simple dan minimalis. Bahasa phyton mudah dipahami dan diimplementasikan sehingga pengembang aplikasi dapat berkonsentrasi pada solusi yang dihadapi.

#### 2. Mudah dipahami

Python memiliki sintaks yang sangat sederhana dari bahasa pemrograman lainnya sehingga dengan mudah dapat dipahami.

#### 3. Gratis dan *Open Source*

Python dapat dengan mudah mendistribusikan salinan perangkat lunak ini, membaca sumber kode, memodifikasi, dan menggunakan bagiannya dan membuat sebuah program baru.

#### 4. Bahasa pemrograman tingkat tinggi

Bahasa tingkat tinggi akan membuat pemrograman komputer yang kompleks menjadi lebih sederhana.

#### 5. *Portable*

Karena sifatnya *Open-Source*, Python telah diubah untuk dapat bekerja pada platform. Semua program Python dapat bekerja pada salah satu platform memerlukan perubahan sama sekali.



## 6. Interpreter

Python mengubah kode sumber menjadi bentuk perantara yang disebut *bytecodes* dan kemudian menerjemahkannya ke dalam bahasa komputer sehingga membuatnya jauh lebih mudah karena tidak perlu melakukan kompilasi program.

## 7. *Object-Oriented*

Python mendukung pemrograman berorientasi prosedur serta pemrograman berorientasi objek. Dalam bahasa *procedure-oriented*, program ini dibangun di sekitar prosedur atau fungsi yang tidak lain adalah potongan program yang dapat digunakan kembali. Dalam bahasa *object-oriented*, program ini dibangun di sekitar objek yang menggabungkan data dan fungsionalitas.

## 8. *Extensible*

Python dapat membaca bagian dari program C atau C++ apabila suatu program membutuhkan sepotong kode penting atau memiliki beberapa bagian dari algoritma dari suatu program.

## 9. *Extensive Libraries*

Standar *library* Python memang sangat besar. Sehingga dapat membantu melakukan berbagai hal yang melibatkan ekspresi regular, pembuatan dokumentasi, pengujian unit, *threading*, *database*, *web browser*, CGI, FTP, *email*, XML, XML\_RPC, HTML, *file WAV*, kriptografi, GUI (*Graphical User Interface*), dan lainnya yang bergantung pada suatu sistem.



kas

on memiliki kombinasi yang tepat dari kinerja dan fitur dalam menulis

program dengan mudah.

## II.4 Django

Django diciptakan pada tahun 2003 di sebuah agensi pers Lawrence, Kansas. Django adalah *web framework* yang menggunakan Python untuk membuat *website*. Tujuannya adalah untuk menulis situs web dinamis yang sangat cepat. Pada tahun 2005, agensi memutuskan untuk mempublikasikan sumber kode Django di lisensi *Berkeley Software Distribution (BSD)*. Pada tahun 2008, *Django Software Foundation* diciptakan untuk mendukung dan memajukan Django. Versi 1.0 dari kerangka ini dirilis beberapa bulan kemudian (Dauzon et al., 2016).

Selama bertahun-tahun, pengembang Django mengadakan konferensi di seluruh dunia untuk dapat memperbaiki *web framework* Django ini menjadi lebih baik. Berikut adalah ringkasan singkat dari perbaikan yang sesuai dengan apa yang pernah digunakan dari kekurangan pada Django dan yang diselesaikan pada perilisannya yaitu (Ravindran, 2015):

1. *Form-handling library* terbaru (Django 0.96)
2. Melepaskan *admin* dari *model* (Django 1.0)
3. Dukungan beberapa *database* (Django 1.2)
4. Mengelola *static files* lebih baik (Django 1.3)
5. Dukungan zona waktu yang lebih baik (Django 1.4)
6. Pengguna *model* yang dapat disesuaikan (Django 1.5)
7. Manajemen transaksi yang lebih baik (Django 1.6)
8. Bangun migrasi *database* (Django 1.7)



Django merupakan sebuah paket *framework* yang kokoh, berikut *libraries* yang terdapat pada *framework* Django dalam melakukan pengembangan *web* (Bennett, 2009):

1. Pemetaan objek yang relasional, yang merupakan sebuah *library* yang tahu seperti apa *database* anda terlihat, seperti apa kode anda terlihat, dan bagaimana untuk menjembatani kesenjangan antara *database* dan kode tanpa menulis *Structured Query Language* (SQL) secara manual dan berulang.
2. Sebuah paket HTTP *libraries* (*Hypertext Transfer Protocol libraries*) yang tahu bagaimana cara mengurai permintaan *web* yang masuk; bagaimana cara memberikannya dalam keadaan standar, format yang mudah digunakan, dan bagaimana cara mengubah hasil dari kode kedalam bentuk yang mudah di tanggapi.
3. *Uniform Resource Locator* (URL) *routing library* yang memungkinkan anda menentukan persis URL yang anda inginkan dan memetakan ke bagian kode yang sesuai.
4. *Validation library* yang membantu anda menampilkan formulir di halaman *web* dan memproses data yang dikirimkan pengguna.
5. Sebuah sistem template yang memungkinkan bahkan *nonprogrammer* dapat menulis *HyperText Markup Language* (HTML) dicampur dengan data yang dihasilkan oleh kode dan dengan jumlah yang tepat dari logika yang ditampilkan.

memiliki kelebihan di antara web framework lainnya yaitu (Anwar, 2018):

*Integration between Components*



Django menyediakan seperangkat komponen yang terintegrasi. Semua komponen telah dikembangkan oleh tim Django sendiri. Django awalnya dikembangkan sebagai *framework in-house* untuk mengelola serangkaian berita yang berorientasi pada situs *web*. Kemudian kodenya dirilis di internet dan tim Django melanjutkannya pengembangan menggunakan model *Open Source*. Oleh sebab itu, komponen Django dirancang untuk *integration*, *reusability* dan *speed* sejak awal.

## 2. *Object-Relational Mapper (ORM, O/RM, and O/R mapping tool)*

Komponen database Django, *Object-Relational Mapper (ORM)*, menyediakan jembatan antara model data dan database *engine*. Ini mendukung satu set besar dalam sistem database, dan beralih dari satu mesin ke mesin lainnya dalam permasalahan perubahan sebuah konfigurasi file. Hal ini memberi fleksibilitas besar bagi pengembang jika menginginkan keputusan dari satu mesin database ke database lainnya.

## 3. *Clean URL Design*

Sistem URL di Django sangat fleksibel dan *powerful*. Ini memungkinkan pendefinisian pola pada URL dan menentukan fungsi Python untuk ditangani pada setiap polanya.

## 4. *Authomatic Admin Interface*

Antarmuka yang disediakan oleh Django membuat pengelolaan data pada aplikasi dengan mudah. Hal ini juga sangat fleksibel dan dapat disesuaikan.



*anced Development Environment*

itu Django menyediakan lingkungan pengembangan yang sangat baik.

Contohnya yaitu pada saat *mode debugging* diaktifkan, Django memberikan pesan kesalahan yang sangat teliti dan terperinci dengan lebih banyak lagi informasi debug. Semua ini membuat proses isolasi dan perbaikan pada *bug* dapat dilakukan dengan sangat mudah.

#### 6. Mendukung Multibahasa

Sistem Django mendukung bahasa pemrograman selain Python dan mengeksekusinya menjadi lebih sederhana sehingga ditampilkan pada antarmuka Django.

### II.5 GeoDjango

GeoDjango adalah modul *contrib* yang disertakan untuk Django yang mengubahnya menjadi kerangka *web* geografis. GeoDjango dibuat sesederhana mungkin dalam membuat aplikasi *web* geografis, seperti layanan berbasis lokasi (docs.djangoprojects.com, 2005).

Ekstensi GeoDjango dibangun berdasarkan kemampuan Django untuk menambahkan dukungan yang lengkap untuk membangun aplikasi *web* geospasial. Secara khusus, fungsi berikut telah ditambahkan ke dalam Django (Westra, 2016)

#### 1. Model

- a. Model Django diperluas untuk mengizinkan menyimpan dan mengambil data geospasial.
- b. *Object Relational Mapping* (ORM) Django diperluas untuk mendukung *query* spasial.

Sebagai fitur geospasial yang dibaca dari database, *Object Relational Mapper* secara otomatis mengubahnya menjadi objek GEOS, menyediakan



metode untuk *query* dan memanipulasi fitur ini dengan cara yang mutakhir.

- d. Model ini dapat mengimpor data dari sumber data vector yang didukung OGR ke *database* GeoDjango.
- e. GeoDjango dapat mengintrospeksi untuk melihat atribut mana yang tersedia dalam sumber data OGR yang diberikan dan secara otomatis mengatur model untuk menyimpan dan mengimpor atribut ini.

## 2. *Template*

Sistem *template* Django diperluas untuk membolehkan tampilan data geospasial menggunakan peta OpenLayers Slippy yang tertanam.

## 3. Antarmuka admin.

Antarmuka admin Django diperluas untuk mengizinkan pengguna membuat dan mengedit data geospasial menggunakan OpenLayers. Data vector ditampilkan diatas peta dasar yang disediakan oleh OpenStreetMap.

## 4. Jarak dan area kalkulator.

Jarak dan area dapat dikonversi antara berbagai unit standar. Misalnya, milimeter, yards, atau mil.

## II.6 MVC Framework

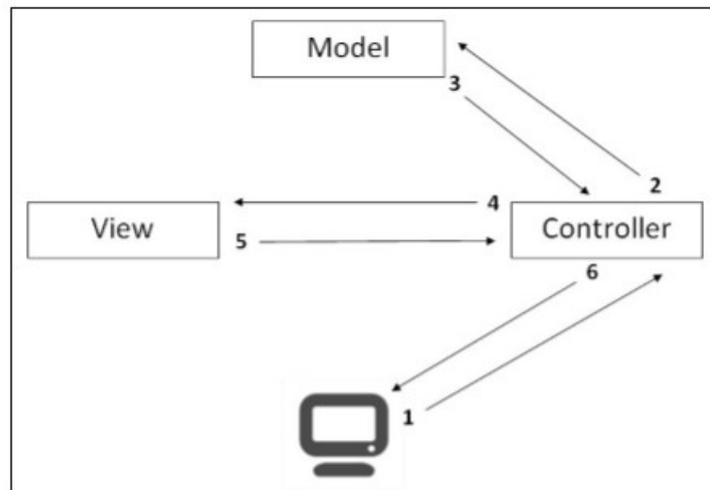
*Framework* adalah perangkat lunak yang mengatur arsitektur aplikasi dan membuat pekerjaan pengembang menjadi lebih mudah. Sebuah framework yang dapat disesuaikan dengan kegunaan yang berbeda. Hal ini juga memberikan alat praktis

membuat pekerjaan *programmer* lebih cepat. Dengan demikian, beberapa yang secara teratur digunakan pada *website* dapat digunakan secara otomatis, administrasi database dan manajemen pengguna (Dauzon et al., 2016).



Pola *Model-View-Controller* (MVC) diciptakan untuk memisahkan logika dari representasi dan memiliki arsitektur internal yang lebih nyata berwujud. MVC mewakili tiga lapisan aplikasi yang direkomendasikan paradigma (Dauzon et al., 2016):

1. *Models*: ini mewakili pengorganisasian data dalam *database*. Dalam kata sederhana, kita dapat mengatakan bahwa setiap *model* mendefinisikan tabel dalam *database* dan hubungan antara *model* lainnya. Berkatnya setiap *bit* data disimpan dalam *database*.
2. *Views*: ini berisi semua informasi yang akan dikirim ke klien. Tampilan yang akan dihasilkan berasal dari dokumen yang berisikan kode HTML.
3. *Controllers*: ini berisi semua tindakan yang dilakukan oleh *server* dan tidak terlihat oleh klien. *Controller* memeriksa apakah pengguna diotentikasi atau dapat menghasilkan kode HTML dari *template*.



**Gambar 2.4** Pola *Model-View-Controller* (MVC) (Sumber: Dauzon et al., 2016).

Berikut ini adalah langkah yang diikuti dalam aplikasi dengan pola MVC (Dauzon et al., 2016):

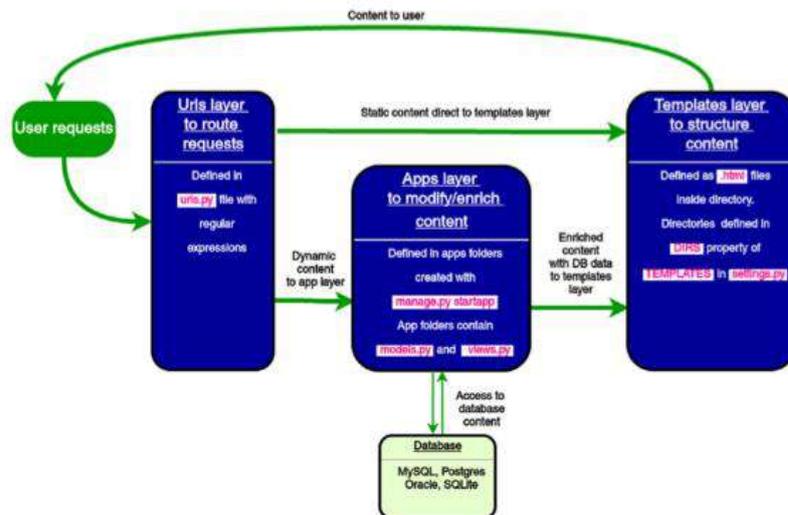


mengirimkan permintaan ke *server* yang meminta untuk menampilkan  
 an.

2. *Controller* menggunakan database melalui *model*. Hal ini dapat membuat, membaca, memperbarui, atau menghapus catatan atau menerapkan logika apapun untuk data yang diperoleh.
3. *Model* mengirimkan data dari database; misalnya, ia mengirimkan daftar produk jika kita memiliki sebuah toko online.
4. *Controller* menyuntikkan data ke dalam tampilan untuk menghasilkannya.
5. *View* mengembalikan isinya bergantung pada data yang diberikan oleh *controller*.
6. *Controller* mengembalikan konten HTML ke klien.

Django menyebut ini adalah arsitektur *Model-View-Template* (MVT). Ada pemisahan kepentingan antara kelas yang berinteraksi dengan *database* (*model*), kelas yang memproses permintaan (*view*), dan sebuah tatanan bahasa untuk presentasi akhir (*template*). Jika dibandingkan dengan pola MVC, *model* dari MVC memiliki kegunaan yang sama dengan arsitektur MVT, *view* pada pola MVC sama dengan *template* pada arsitektur MVT, sedangkan *controller* adalah *framework* itu sendiri yang memproses permintaan HTTP yang masuk, dan merutekan ke fungsi tampilan yang benar disebut *view* pada arsitektur MVT (Ravindran, 2015).





**Gambar 2.5** Alur kerja Django untuk URL, template, dan aplikasi (Sumber: Rubio, 2017).

## II.7 Paket *Contrib*

Paket *contrib* adalah bagian dari Django yang mengandung beberapa aplikasi yang sangat berguna untuk disertakan pada Django. Aplikasi yang disertakan memberikan seperangkat fitur yang mengesankan, termasuk beberapa yang akan digunakan dalam aplikasi ini (Ahmed, 2016):

1. *Admin* adalah *Content Management System* (CMS) fitur lengkap yang dapat digunakan untuk mengelola isi dari situs Django. Aplikasi *admin* merupakan alasan penting untuk popularitas Django. Ini berguna menyediakan sebuah antarmuka untuk situs administrator pada umumnya.
2. *Auth* menyediakan pendaftaran dan autentikasi pengguna tanpa mengharuskan

melakukan apapun. Modul ini memungkinkan pengguna untuk mendaftar, login, dan mengelola profil mereka pada aplikasi.



*Admin* yang berada pada aplikasi *contrib* GeoDjango disebut *GeographicAdmin*. *GeographicAdmin* adalah sebuah contoh proyek yang menyoroti *admin* dan *databrowse* Django dengan fitur spasial GeoDjango (yang tertanam peta *Openlayers*) yang tersedia (code.google.com, 2006).

Adapun paket *contrib* yang digunakan untuk membuat aplikasi *web-GIS* menggunakan *framework* GeoDjango, yaitu:

1. GDAL
2. Psycopg2
3. Jsonfield
4. Django
5. Django-geojson
6. Django-leaflet
7. Django-shapes
8. Django-form-utils
9. Django-crispy-forms
10. Django-floppyforms
11. Django-widget-tweaks

## **II.8 Spasial Database**

*Database* yang diaktifkan spasialnya adalah alat yang sangat ampuh untuk bekerja dengan data geospasial. Dengan menggunakan indeks spasial dan optimasi lainnya,

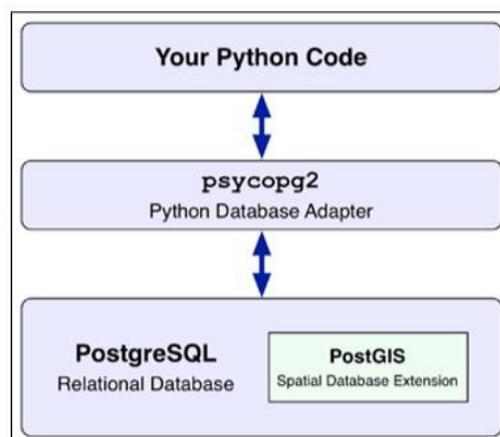
spasial dapat dengan cepat melakukan pekerjaan dengan jenis operasi



seperti ini dan dapat menskalakan untuk mendukung sejumlah besar data yang tidak layak menggunakan skema penyimpanan data lainnya (Westra, 2016).

Salah satu karakteristik yang mendefinisikan *database* spasial adalah kemampuan untuk membuat dan menggunakan indeks “spasial” untuk mempercepat pencarian berbasis geometri. Indeks ini digunakan untuk melakukan operasi spasial, seperti mengidentifikasi semua fitur yang ada dalam kotak imajiner yang diberikan, mengidentifikasi semua fitur dalam jarak tertentu dari suatu titik, atau mengidentifikasi semua fitur yang bersinggungan dengan polygon tertentu (Westra, 2016).

PostGIS adalah salah satu *database* geospasial yang paling populer dan kuat, dan bersifat *open source* dan tersedia secara bebas. PostGIS merupakan perluasan ke sistem *database* relasional PostgreSQL. Untuk dapat menggunakan PostGIS dari program Python, maka harus memasang dan mengatur PostgreSQL, kemudian memasang ekstensi PostGIS dan kemudian memasang adaptor *database* psycopg2 untuk Python (Westra, 2016).



**Gambar 2.6** Alur kerja PostGIS (Sumber: Westra, 2016)



PostGIS memberikan akses untuk menyimpan dan melakukan *query* terhadap berbagai jenis data spasial, termasuk titik, garis, poligon, dan koleksi geometri. PostGIS menyediakan dua jenis bidang spasial yang dapat digunakan untuk menyimpan data spasial (Westra, 2016):

1. Bidang geometri menahan data spasial yang diasumsikan berada dalam sistem koordinat yang diproyeksikan. Semua perhitungan dan *query* untuk bidang geometri mengasumsikan bahwa data spasial telah diproyeksikan kedalam bidang *cartesian* datar. Hal ini membuat perhitungan menjadi lebih sederhana, tetapi itu hanya akan bekerja jika data spasial berada dalam sistem koordinat yang diproyeksikan.
2. Bidang geografi menyimpan data spasial yang menggunakan koordinat geodetik (tidak diproyeksikan). Perhitungan dan *query* terhadap bidang geografi mengasumsikan bahwa data berada di sudut unit (yaitu nilai lintang dan bujur), menggunakan matematika yang mutakhir untuk menghitung panjang dan area menggunakan model *spheroid* dari bumi.

Karena matematika yang terlibat jauh lebih rumit, tidak semua fungsi spasial tersedia untuk bidang geografi, dan operasi sering memakan waktu yang lebih lama. Namun, bidang geografi jauh lebih mudah digunakan jika data spasial menggunakan sistem koordinat yang tidak diproyeksikan seperti WGS84 (Westra, 2016)

## II.9 Lingkungan pengembangan proyek Django



pat menggunakan Django, *software* yang perlu dipasang yaitu (Dauzon et  
:

1. Python, untuk dapat berinteraksi dengan *framework* Django.
2. *Setuptools* adalah modul yang menyederhanakan instalasi modul Python eksternal. Namun, tidak dapat mengatur pelepasan modul.
3. PIP memperluas kemungkinan *setuptools* dengan menghapus paket, menggunakan sintaks yang lebih mudah, dan memberikan manfaat lainnya.
4. Django, yang mana dapat dipasang melalui PIP.

### II.9.1 Command Line

*Command line* merupakan baris perintah yang kuat yang hanya menampilkan teks pada komputer. Sebagai pengembang, *command line* akan digunakan secara luas untuk menginstal dan mengkonfigurasi setiap proyek Django. Sementara ada banyak perintah yang mungkin bisa digunakan, dalam praktiknya ada enam perintah yang paling sering digunakan dalam pengembangan Django (Vincent, 2020):

1. `cd` (kembali ke direktori sebelumnya)
2. `cd ..` (membuka direktori selanjutnya)
3. `ls` (daftar file dalam direktori saat ini)
4. `pwd` (cetak direktori yang bekerja)
5. `mkdir` (membuat direktori)
6. `touch` (membuat file baru)

### II.9.2 Virtual Environment

*Environment* adalah bagian yang tak terpisahkan dari pemrograman Python. Ini akan wadah terisolasi yang berisi semua komponen terkait pada perangkat lunak untuk proyek tertentu. Ini merupakan sesuatu yang penting karena secara



standar perangkat lunak seperti Python dan Django dipasang di direktori yang sama. Hal ini menyebabkan masalah ketika melakukan beberapa pekerjaan proyek di komputer yang sama (Vincent, 2020).

Tanpa *VirtualEnv*, masih dapat melanjutkan pemasangan Django dan paket Python lainnya menggunakan pip, tapi memiliki masalah pada saat semua paket dipasang dibawah instalasi Python global. Awalnya ini dapat tampak nyaman, karena hanya perlu memasang paket sekali dalam instalasi Python global. Untuk mengatasi masalah ini yaitu dengan pilihan pertama memerlukan pekerjaan tambahan, sementara pilihan kedua mengharuskan untuk mengembangkan versi Django yang ketinggalan ataupun paket Python lainnya yang ketinggalan. Dengan menggunakan lingkungan *virtual* Python masalah seperti ini dapat dihindari, karena setiap proyek dapat menjalankan versi Django sendiri di dalam lingkungan yang di isolasi (Rubio, 2017).

Bagian terpenting dari *virtual environment* adalah *executable* yang berada di direktori bin. Jika anda menggunakan salah satu *executable* ini, seperti pip, *easy install*, atau *wheel*. Ini di eksekusi dibawah konteks lingkungan *virtual* Python. sebagai contoh, pip yang berada di folder bin memasang paket untuk *virtual environment*. Demikian pula, sebuah aplikasi yang berjalan menggunakan *executable* Python yang berada pada folder bin hanya mampu memuat paket yang dipasang pada lingkungan *virtual* Python. ini merupakan peran “*sandbox*” yang

butkan sebelumnya (Rubio, 2017).



*VirtualEnv* memiliki mekanisme untuk memuat *Virtual Environment* sehingga jika mengeksekusi pip, Python, atau *executable* lainnya, maka *executable* yang jalan berasal dari lingkungan *virtual Python* yang telah diaktifkan (bukan dari *default Python* global). Berikut contoh perintah yang dijalankan dari sistem Linux (Rubio, 2017).

**Command line:**

```
-----  
[user@~]$ source ./bin/activate  
[(virtualenv)user@~]$  
-----
```

Dari perintah diatas, setelah diaktifkan *executable* yang dipanggil, *command prompt* menambahkan nama *virtual environment* diantara tanda kurung. Ini berarti *executable* di dalam direktori bin dari lingkungan *virtual Python* digunakan diluar dari Python global yang terpasang. Untuk keluar dari lingkungan *virtual Python* cukup ketik “*deactivate*” dan *executable* dari Python global yang berjalan (Rubio, 2017).

**II.9.3 Python Installation Packages**

*Python Installation Packages* (PIP) adalah paket manajer dari Python yang membuatnya mudah untuk memasang paket Python dari repositori Python pihak ketiga resmi yaitu PyPi. PIP Python telah dipasang secara permanen pada Python-2 versi 2.7.9 atau lebih tinggi dan Python-3 versi 3.4 atau lebih tinggi. Pada sistem operasi Linux, lokasi *default* untuk PIP Python sama dengan Python *executable*.

Misalnya, jika pengguna memiliki /usr/bin/python, maka PIP Python harusnya di /usr/bin/pip. Pada sistem operasi Windows, pip.exe terdapat di





```
[user@~]$ pip install /home/Downloads/Django-1.11.tar.gz
Processing /home/Downloads/Django-1.11.tar.gz
Collecting pytz (from Django==1.11)
  Using cached pytz-2017.2-py2.py3-none-any.whl
Building wheels for collected packages: Django
  Running setup.py bdist_wheel for Django ... done
  Stored in directory: /home/ubuntu/.cache/pip/wheels/56/bf/24/
f44162e115f4fe0cfeb4b0ae99b570fb55a741a8d090c9894d
Successfully built Django
Installing collected packages: pytz, Django
Successfully installed Django-1.11 pytz-2017.2
```

**Gambar 2.8** Pemasangan Django menggunakan file yang berformat tar.gz

(Sumber: Rubio, 2017)

Nama proyek Django dapat terdiri dari angka, huruf, atau garis bawah. Nama proyek tidak dapat dimulai dengan angka, itu hanya dapat dimulai dengan huruf atau garis bawah. Selain itu, karakter khusus, dan spasi tidak diperbolehkan dimana saja dalam nama proyek, terutama karena nama proyek Django sebagai konvensi penamaan untuk direktori dan paket Python (Rubio, 2017).

Gunakan skrip standar Django-admin untuk membuat sebuah proyek bernama geodjango (docs.djangoproject.com, 2005):

**Command line:**

```
-----
$ Django-admin startproject geodjango
-----
```

Perintah ini membuat folder geodjango di dalam folder yang dimana tempat menjalankan perintah (Dauzon et al., 2016). Jika melihat tata letak direktori, direktori proyek Django bagian luar sebagai BASE\_DIR, yang meliputi file manage.py dan sub-direktori lain yang berdasarkan nama proyek. Dan isi dari sub-

terdapat file `__init__.py`, `settings.py`, `urls.py`, dan `wsgi.py` sebagai `T_DIR` (Rubio, 2017).



```

+<BASE_DIR_project_name>
|
+----manage.py
|
+----+<PROJECT_DIR_project_name>
      |
      +- __init__.py
      +- settings.py
      +- urls.py
      +- wsgi.py

```

**Gambar 2.9** Struktur proyek Django (Rubio, 2017)

Berikut tujuan setiap file yang berada di struktur proyek Django (Rubio, 2017):

1. Manage.py bertujuan untuk menjalankan tugas yang spesifik dari proyek. Sama seperti Django-admin yang digunakan untuk menjalankan sistem secara meluas dari tugas Django.
2. \_\_init\_\_.py merupakan file Python yang secara generik digunakan di hampir semua aplikasi Python.
3. Settings.py berisi pengaturan konfigurasi untuk proyek Django.
4. Urls.py berisi pola URL untuk proyek Django.
5. Wsgi.py berisi properti konfigurasi WSGI untuk proyek Django. WSGI adalah pendekatan yang direkomendasikan untuk menyebarkan aplikasi Django pada pemroduksian (yaitu untuk umum). Tidak perlu mengatur WSGI untuk mengembangkan aplikasi Django.



## II.10 Geospasial libraries

### II.10.1 GEOS

GEOS adalah sebuah pustaka C++ untuk melakukan operasi geometric, dan merupakan representasi geometri internal yang secara *default* digunakan oleh GeoDjango (dibalik geometri “*Lazy*”). Secara khusus, C API *library* disebut (seperti `libgeos_c.so`) langsung dari Python menggunakan ctypes (docs.djangoprojects.com, 2005).

### II.10.2 PROJ.4

PROJ.4 adalah singkatan untuk versi 4 dari PROJ library. PROJ awalnya ditulis oleh *US Geological Survey* untuk menangani proyeksi peta dan telah banyak digunakan dalam perangkat lunak geospasial selama bertahun-tahun. PROJ adalah kelas transformasi kartografi yang digunakan untuk mengkonversi koordinat geografis (yaitu nilai lintang dan bujur) ke koordinat kartografi (nilai (x,y), secara *default* dalam meter) dan sebaliknya (Westra, 2016).

### II.10.3 GDAL

GDAL (*Geospatial Data Abstraction Library*), awalnya hanya *library* untuk bekerja dengan data geospasial yang berformat raster, sedangkan *library* OGR bekerja dengan data geospasial yang berformat vektor. Namun, kedua *library* ini digabung dengan nama gabungan GDAL (Westra, 2016).

