

# Flask MySQL Boilerplate

Setup and Troubleshoot

# Anggota

- Suparman (31)
- Syanindita Nandira
- Maharani (32)
- Tantowi Jauhari(33)
- Venny Handayani (34)
- Yosua Tito Sumbogo (35)
- Zaky (36)



# Clone Flask

1. Run ***Git Bash.exe***
2. Lalu jalankan perintah  
-> *git clone*  
*https://github.com/fromzeroedu/flask-mysql-boilerplate*

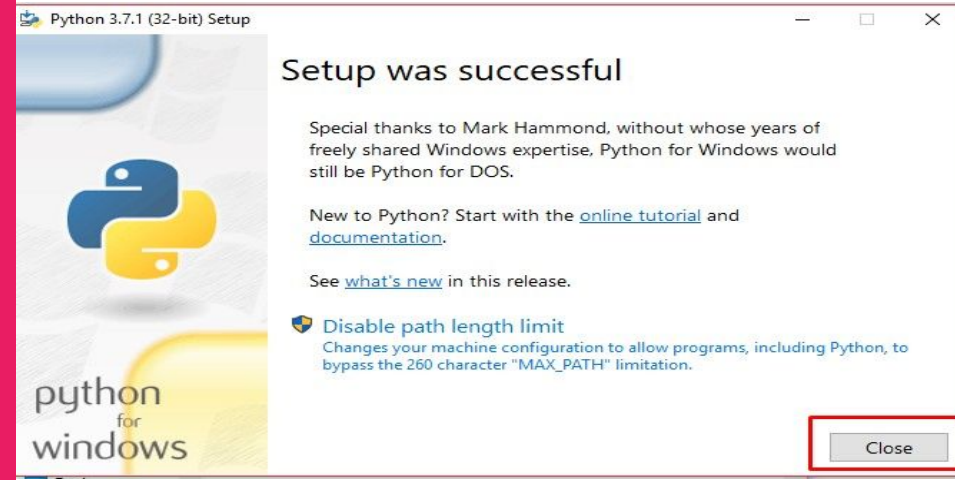
“Git clone” berfungsi untuk menyalin melalui link repository online ke direktori local kita

```
zakva@YXG MINGW64 ~  
$ git clone https://github.com/fromzeroedu/flask-mysql-boilerplate  
Cloning into 'flask-mysql-boilerplate'...  
remote: Enumerating objects: 71, done.  
remote: Total 71 (delta 0), reused 0 (delta 0), pack-reused 71  
Unpacking objects: 100% (71/71), done.
```

— — —

# Install Python

1. Klik **Install Now**
2. Tunggu proses hingga selesai
3. Klik **Close**



# Create database & Users



---

# Create database

Login MySQL menggunakan root user via cmd

1. Ketik “ssh root@(ip address)” kemudian masukan password root (Gambar 1)
2. Masuk ke MySQL, ketik “mysql -u root -p” kemudian masukan password MySQL (Gambar 1)
3. Buat database dengan cara, ketik “CREATE DATABASE counter;” (Gambar 2)

```
C:\Users\Azizah>ssh root@103.101.224.205
root@103.101.224.205's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-109-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage
New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Oct 31 17:48:23 2018 from 152.118.37.246
root@dbcloudsore7:~#mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 104
Server version: 5.7.24-0ubuntu0.16.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Gambar 1

```
mysql> CREATE DATABASE counter;
Query OK, 1 row affected (0.00 sec)
```

Gambar 2

# Create users & Give previlages

Setelah database 'counter' dibuat, selanjutnya siapkan user

1. Buat user, ketik "CREATE USER 'counter\_user'@'%' IDENTIFIED BY 'counter\_password';" pada cmd
2. Berikan previlage pada 'couter\_user' dengan mengetik "GRANT ALL PRIVILEGES ON counter.\* TO 'counter\_user'@'%'";
3. Kemudian reload MySQL dengan mengetik "FLUSH PRIVILEGES;"

```
mysql> CREATE USER 'counter_user'@'%' IDENTIFIED BY 'counter_password';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> GRANT ALL PRIVILEGES ON counter.* TO 'counter_user'@'%' ;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)
```

# Create venv

Virtual Environment



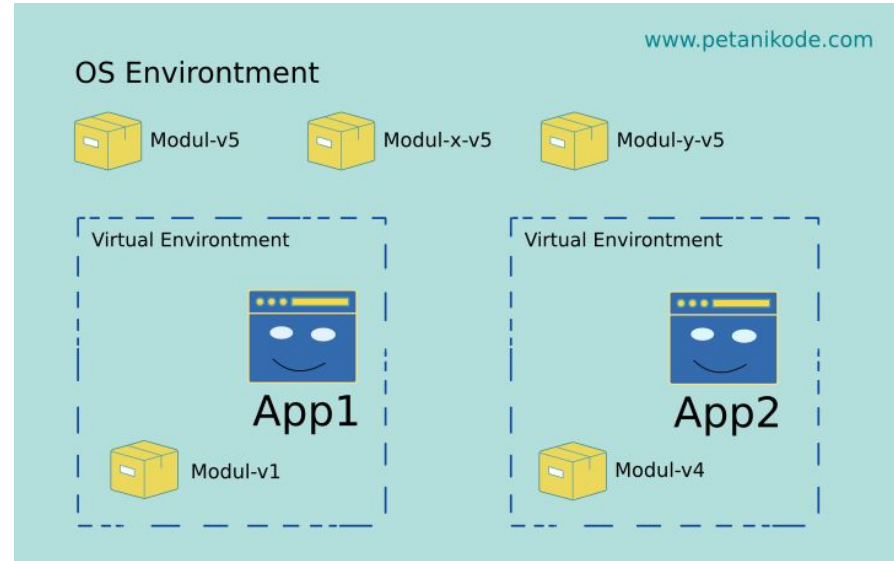
---



# Apa itu venv?

Virtualenv adalah tools untuk membuat lingkungan python virtual yang terisolasi. Terisolasi artinya tertutup dan tidak bisa diakses dari dunia luar.

Program Python yang berjalan di dalam virtualenv memiliki modul-modulnya sendiri dan program dari luar tidak bisa mengaksesnya. Sedangkan program Python yang berjalan tanpa virtualenv hanya bisa menggunakan modul-modul global saja, yang berada di `/usr/lib/python2.7/site-packages`.



# Create venv

Pertama-tama anda harus mengganti directory anda sesuai dengan dimana anda menyimpan data 'flask-mysql-boilerplate' nya dengan cara ketik "cd (directory anda)" pada cmd  
Setelah itu ketik "py -m venv venv" atau "python3 -m venv venv" untuk membuat virtual environmentnya.

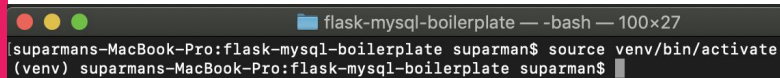
```
C:\Users\Azizah>cd C:\Users\Azizah\Documents\Towi\Cloud\Git\Git Flask\flask-mysql-boilerplate
C:\Users\Azizah\Documents\Towi\Cloud\Git\Git Flask\flask-mysql-boilerplate>py -m venv venv
C:\Users\Azizah\Documents\Towi\Cloud\Git\Git Flask\flask-mysql-boilerplate>
```

# Activate venv

Aktifkan venv dengan perintah

di linux/mac :  
`source venv/bin/activate`

di windows :  
`venv/Scripts/activate`



A terminal window titled "flask-mysql-boilerplate — -bash — 100x27". The prompt is "suparmans-MacBook-Pro:flask-mysql-boilerplate superman\$". The command "source venv/bin/activate" has been entered, and the prompt has changed to "(venv) suparmans-MacBook-Pro:flask-mysql-boilerplate superman\$".

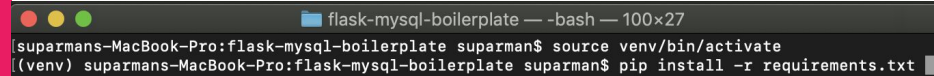
```
flask-mysql-boilerplate — -bash — 100x27
suparmans-MacBook-Pro:flask-mysql-boilerplate superman$ source venv/bin/activate
(venv) suparmans-MacBook-Pro:flask-mysql-boilerplate superman$
```

# Install Packages

Setiap program/ aplikasi tidak mungkin berdiri sendiri. Mereka pasti memerlukan aplikasi/ library lain (dependencies).

Install dependencies yang diperlukan dengan perintah

```
Pip install -r requirement.txt
```



```
flask-mysql-boilerplate — -bash — 100x27
suparmans-MacBook-Pro:flask-mysql-boilerplate superman$ source venv/bin/activate
((venv) suparmans-MacBook-Pro:flask-mysql-boilerplate superman$ pip install -r requirements.txt
```

# Install Packages

result

```
flask-mysql-boilerplate — -bash — 100x27
Requirement already satisfied: Flask-SQLAlchemy==2.3.2 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 11)) (2.3.2)
Requirement already satisfied: itsdangerous==0.24 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 12)) (0.24)
Requirement already satisfied: Jinja2==2.10 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 13)) (2.10)
Requirement already satisfied: Mako==1.0.7 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 14)) (1.0.7)
Requirement already satisfied: MarkupSafe==1.0 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 15)) (1.0)
Requirement already satisfied: PyMySQL==0.8.1 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 16)) (0.8.1)
Requirement already satisfied: python-dateutil==2.7.3 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 17)) (2.7.3)
Requirement already satisfied: python-dotenv==0.8.2 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 18)) (0.8.2)
Requirement already satisfied: python-editor==1.0.3 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 19)) (1.0.3)
Requirement already satisfied: six==1.11.0 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 20)) (1.11.0)
Requirement already satisfied: SQLAlchemy==1.2.7 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 21)) (1.2.7)
Requirement already satisfied: Werkzeug==0.14.1 in ./venv/lib/python3.7/site-packages (from -r requirements.txt (line 22)) (0.14.1)
You are using pip version 10.0.1, however version 18.1 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
(venv) suparmans-MacBook-Pro:flask-mysql-boilerplate suparman$
```

# Change setting.py

PADA TAHAP INI UNTUK MENYAMBUNGAN APLIKASI KE DATABASE YANG TELAH DIBUAT (COUNTER) PADA VIRTUAL SERVER MAKA LAKUKAN LANGKAH BERIKUT.

— — —

# settings.py

— — —

UBAH KONFIGURASI SETTINGS.PY PADA REPOSITORY FLASK-MYSQL-BOILERPLATE

```
1  import os
2
3  SECRET_KEY=os.environ['SECRET_KEY']
4  DB_USERNAME=os.environ['DB_USERNAME']
5  DB_PASSWORD=os.environ['DB_PASSWORD']
6  DB_HOST=os.environ['DB_HOST']
7  DATABASE_NAME=os.environ['DATABASE_NAME']
8  DB_URI = "mysql+pymysql://%s:%s@%s:3306/%s" % (DB_USERNAME, DB_PASSWORD, DB_HOST, DATABASE_NAME)
9  SQLALCHEMY_DATABASE_URI = DB_URI
10 SQLALCHEMY_TRACK_MODIFICATIONS = True
11 MYSQL_ROOT_PASSWORD=os.environ['MYSQL_ROOT_PASSWORD']
```

# Settings.py (cont...)

— — —

SESUAIKAN DB\_USERNAME, PASSWORD, HOST, DAN DATABASE\_NAME DENGAN KONFIGURASI DATABASE PADA MYSQL SERVER

```
1 import os
2
3 SECRET_KEY=os.environ['SECRET_KEY']
4 DB_USERNAME='counter_user'
5 DB_PASSWORD='counter_password'
6 DB_HOST='103.101.224.205'
7 DATABASE_NAME='counter'
8 DB_URI = "mysql+pymysql://%s:%s@%s:3306/%s" % (DB_USERNAME, DB_PASSWORD, DB_HOST, DATABASE_NAME)
9 SQLALCHEMY_DATABASE_URI = DB_URI
10 SQLALCHEMY_TRACK_MODIFICATIONS = True
11 MYSQL_ROOT_PASSWORD=os.environ['MYSQL_ROOT_PASSWORD']
12
```



# flask init db

Inisiasi database

SETELAH KONFIGURASI PADA FILE SETTINGS.PY KETIK PERINTAH  
“flask db init” UNTUK  
MENGINISIALISASI MIGRASI DATABASE.

— — —

# flask db init

---

```
(venv) D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate>flask db init
Creating directory D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations ... done
Creating directory D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations\versions ... done
Generating D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations\alembic.ini ... done
Generating D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations\env.py ... done
Generating D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations\README ... done
Generating D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations\script.py.mako ... done
Please edit configuration/connection/logging settings in 'D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations\alembic.ini' before proceeding.
```

# Flask db migrate

migrasi database



---

# Flask db migrate (cont...)

— — —

Saat aplikasi menjadi semakin besar, mungkin kita akan mengubah isinnya atau untuk menambah data baru dan juga mengubah serta menghapus data yang sudah ada (ubah skema database).

Tahap ini adalah melakukan migrasi database akan tetapi melalui skrip yang akan membuat perubahan schema sedemikian rupa sehingga database tidak perlu dibuat ulang dari awal. Caranya dengan mengetikan perintah

```
flask db migrate
```

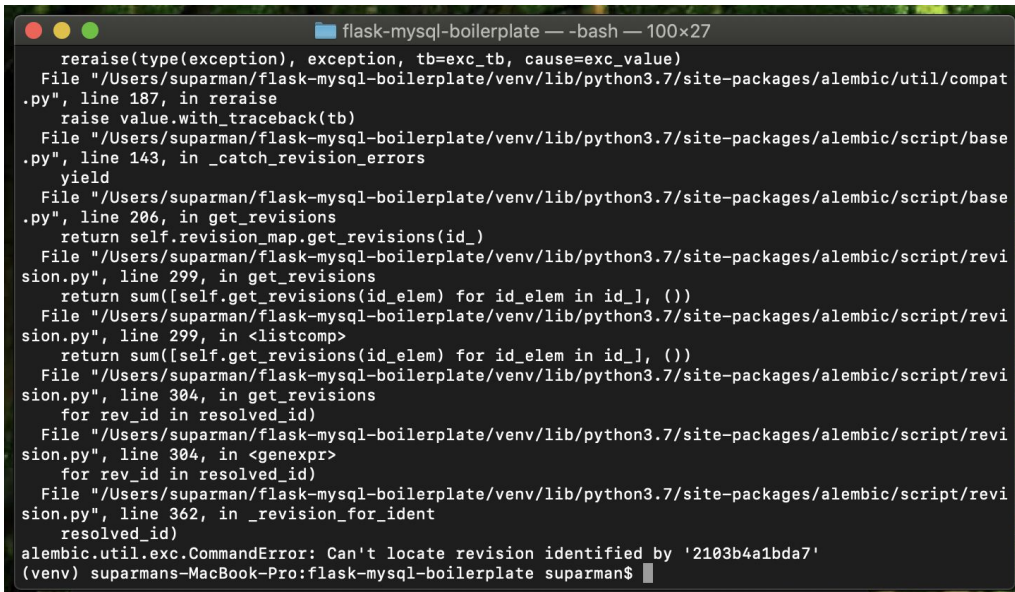
Perintah ini tidak mengubah apapun ke database, ia hanya membuatkan skrip migration

```
(venv) D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate>flask db migrate
INFO [alembic.runtime.migration] Context impl MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
INFO [alembic.autogenerate.compare] Detected added table 'counter'
Generating D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate\migrations\versions\2103b4a1bda7_.py ... done
```

# Flask db migrate (cont...)

— — —

Jika terjadi error pada saat migrate dan upgrade kemungkinan besar itu karena alembic (migration framework) telah menyimpan version history (version\_num) pada database yang sudah di migrasi di awal.

A terminal window titled 'flask-mysql-boilerplate' with a '-bash' prompt and a window size of '100x27'. The terminal displays a stack trace for an 'alembic.util.exc.CommandError'. The error message is 'Can't locate revision identified by '2103b4a1bda7''. The stack trace starts from the bottom with the error message and goes upwards through several files in the alembic package, including 'script/base.py', 'script/revision.py', and 'util/compat.py'. The terminal ends with the prompt '(venv) suparmans-MacBook-Pro:flask-mysql-boilerplate suparman\$'.

```
flask-mysql-boilerplate — -bash — 100x27
  reraise(type(exception), exception, tb=exc_tb, cause=exc_value)
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/util/compat.py", line 187, in reraise
    raise value.with_traceback(tb)
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/script/base.py", line 143, in _catch_revision_errors
    yield
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/script/base.py", line 206, in get_revisions
    return self.revision_map.get_revisions(id_)
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/script/revision.py", line 299, in get_revisions
    return sum([self.get_revisions(id_elem) for id_elem in id_l], ())
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/script/revision.py", line 299, in <listcomp>
    return sum([self.get_revisions(id_elem) for id_elem in id_l], ())
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/script/revision.py", line 304, in get_revisions
    for rev_id in resolved_id)
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/script/revision.py", line 304, in <genexpr>
    for rev_id in resolved_id)
  File "/Users/suparman/flask-mysql-boilerplate/venv/lib/python3.7/site-packages/alembic/script/revision.py", line 362, in _revision_for_id
    resolved_id)
alembic.util.exc.CommandError: Can't locate revision identified by '2103b4a1bda7'
(venv) suparmans-MacBook-Pro:flask-mysql-boilerplate suparman$
```

# Flask db migrate (cont...)

---

Solusinya hapus table  
*alembic\_version* yang memuat  
version\_num dengan perintah

`drop table alembic_version`

pada mySQL

Kemudian lakukan flask init db dan  
flask db migrate kembali

```
[mysql> select * from alembic_version;
+-----+
| version_num |
+-----+
| 2103b4a1bda7 |
+-----+
1 row in set (0.00 sec)

[mysql> drop table alembic_version
-> ;
Query OK, 0 rows affected (0.14 sec)
```

# Flask db upgrade

eksekusi migrasi database



---

# Flask db upgrade (cont...)

— — —

Tahap ini untuk melakukan perubahan ke dalam database sesuai dengan apa yang sudah di skemakan pada step sebelumnya yakni migrasi.

Hal ini dilakukan dengan perintah

`flask db upgrade`

```
(venv) D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate>flask db upgrade
INFO [alembic.runtime.migration] Context impl MySQLImpl.
INFO [alembic.runtime.migration] Will assume non-transactional DDL.
INFO [alembic.runtime.migration] Running upgrade -> 2103b4a1bda7, empty message
```



# Flask run

Tahap ini buat sebuah script “flask run” pada cmd untuk menjalankan web server

```
(venv) D:\KULIAH\Digitalent 2018\flask-mysql-boilerplate>flask run
* Serving Flask app "manage.py" (lazy loading)
* Environment: development
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 211-867-801
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

— — —

# Open browser

Tahap terakhir adalah membuka browser dan tulis <http://localhost:5000/>, maka akan menampilkan sebuah halaman dengan tampilan seperti gambar berikut :



**Thank You**