



DIGITAL TALENT SCHOLARSHIP 2019

Big Data Analytics



Project 3: Konfigurasi RESTful API untuk Web App & Mobile App + Spark Streaming

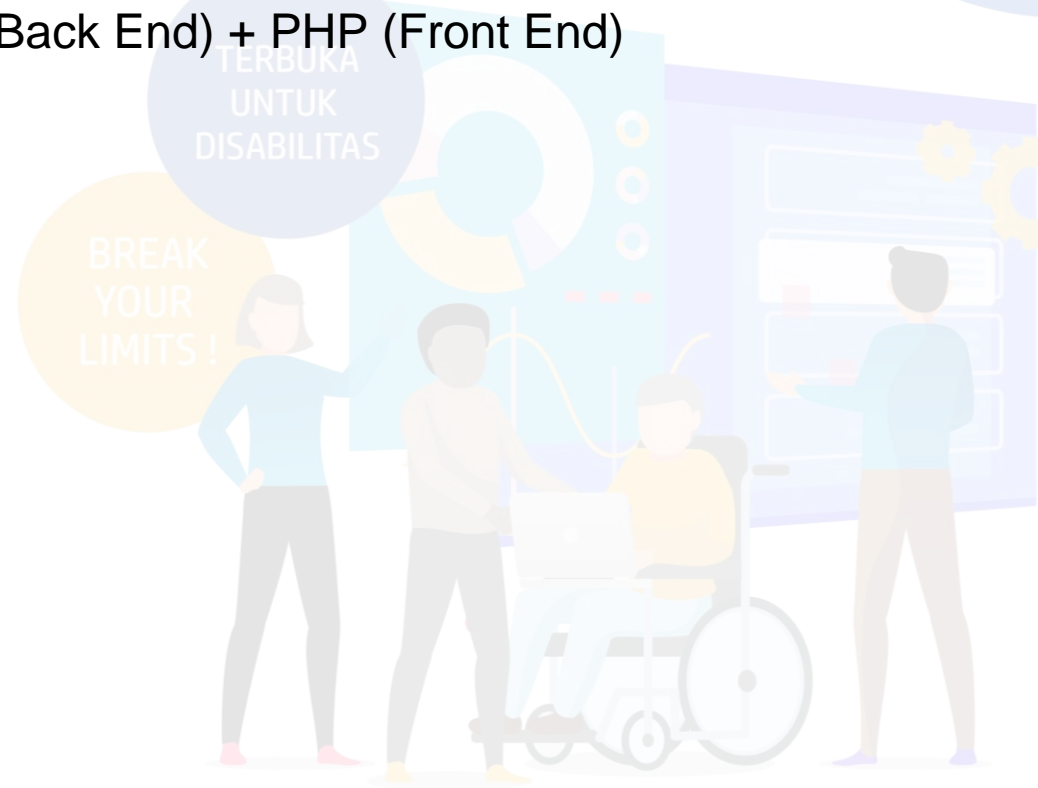
Oleh: Imam Cholissodin | imamcs@ub.ac.id, Putra Pandu Adikara, Sufia Adha Putri

Asisten: Guedho, Sukma, Anshori, Aang dan Gusti

Fakultas Ilmu Komputer (Filkom) Universitas Brawijaya (UB)

Pokok Pembahasan

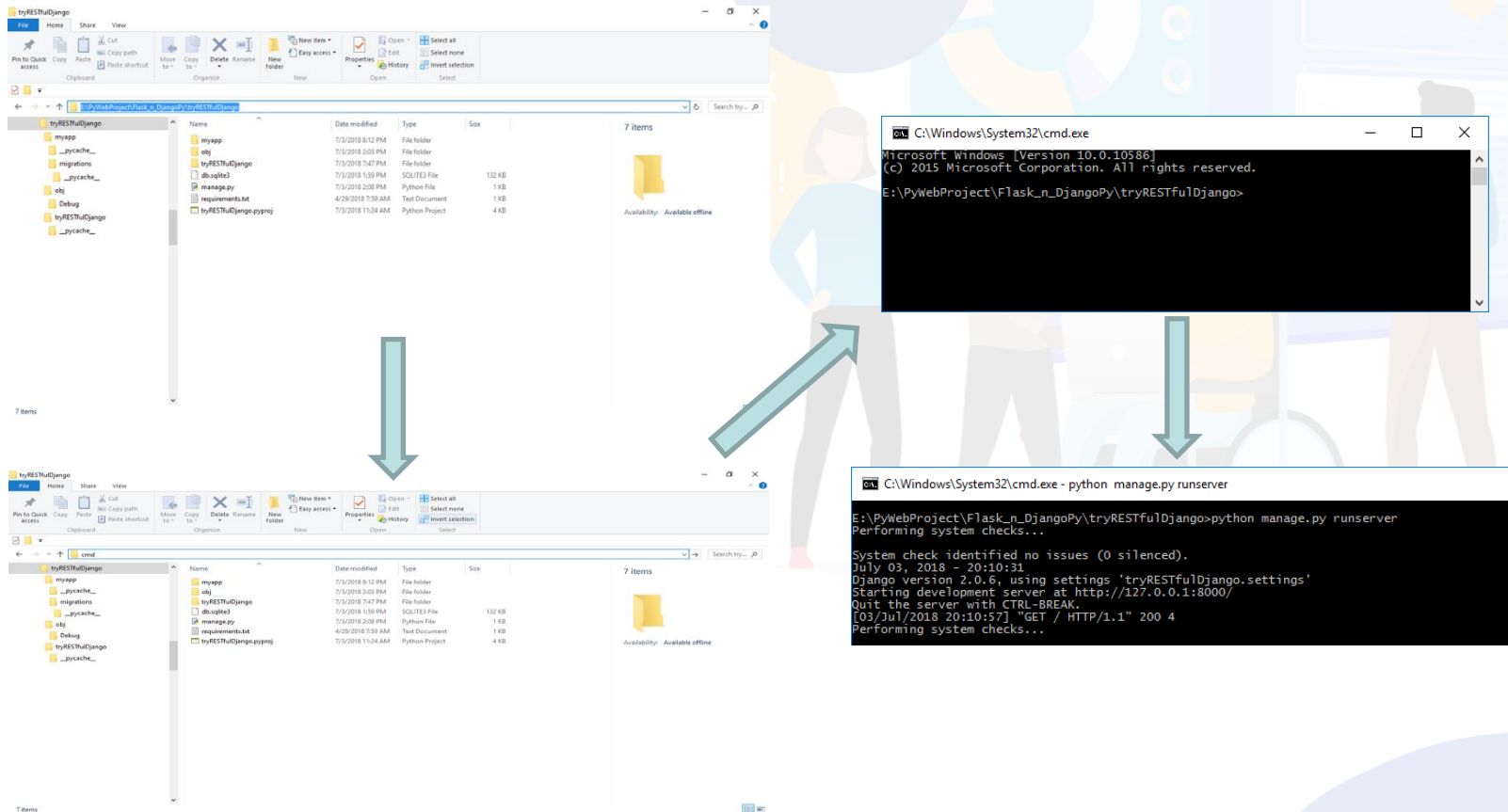
- Mobile App dgn Ionic/Flutter dan Web App dgn Django (dari Project 2)
- Contoh Project: Python Django (Back End) + PHP (Front End)



Tahapan Koding (Dari Contoh Fix di Local Komputer): Python Django (Back End) + Other

Python Django (Web)

- Tahapan Koding RESRful di Python Django (Web):
Link Full source code “<http://bit.ly/2U8q3XI>”
 - Buka cmd dari folder project “tryRESTfulDjango”, pada path bagian yang diblok warna biru ketik “cmd” tanpa quote, lalu tekan enter, ketik “python manage.py runserver”



Python Django (Web)

- Tahapan Koding RESRful di Python Django (Web):
 - Coba ketik “curl http://localhost:8000/Tesla%20Model%20SS” terlihat bahwa django GET akses property mobil dengan nama “Tesla Model SS” dengan mengembalikan dua property

```
C:\Windows\system32\cmd.exe

C:\Users\Imacho>curl http://localhost:8000/Tesla%20Model%20SS
[{"Car": "Tesla Model SS", "Top Speed": 155}]
C:\Users\Imacho>
```

- Coba ketik “curl -d "{\"car_name\":\"Tesla Model S4\",\"top_speed\":\"155\"}\" -H \"Content-Type: application/json\" -X POST http://localhost:8000/car” atau “curl -d "{\"car_name\":\"Tesla Model S4\",\"top_speed\":55}\" -H \"Content-Type: application/json\" -X POST http://localhost:8000/car” untuk menambah data mobil ke database “db.sqlite3”

```
C:\Windows\system32\cmd.exe

C:\Users\Imacho>curl -d "{\"car_name\":\"Tesla Model S4\",\"top_speed\":\"155\"}\" -H \"Content-Type: application/json\" -X POST http://localhost:8000/car
[{"Success": "Car added successfully!"}]
C:\Users\Imacho>curl -d "{\"car_name\":\"Tesla Model S4\",\"top_speed\":55}\" -H \"Content-Type: application/json\" -X POST http://localhost:8000/car
[{"Success": "Car added successfully!"}]
C:\Users\Imacho>
```

Python Django (Web)

- Tahapan Koding RESRful di Python Django (Web):
 - Cek dengan Aplikasi sqlite untuk melihat hasil update data mobil:

The screenshot shows the SQLite Forensic Explorer application. The left sidebar displays a tree view of the database structure, including tables like 'django_migrations', 'sqlite_sequence', 'auth_group', and 'myapp_car' (10). The main window is divided into two panes: 'Tabular' and 'Hex'. The 'Tabular' pane shows a table with columns 'id', 'name', and 'top_speed'. The 'Hex' pane shows the raw hex data for the selected row.

| id | name | top_speed |
|----|----------------|-----------|
| 1 | Tesla Model SS | 155 |
| 2 | Ferrari 488 | 202 |
| 3 | Bugatti Chiron | 250 |
| 4 | Tesla Model S3 | 155 |
| 5 | Tesla Model S4 | 155 |
| 6 | Tesla Model S4 | 155 |
| 7 | Tesla Model S4 | 155 |
| 8 | Tesla Model S4 | 155 |
| 9 | Tesla Model S4 | 55 |
| 10 | Tesla Model S4 | 55 |

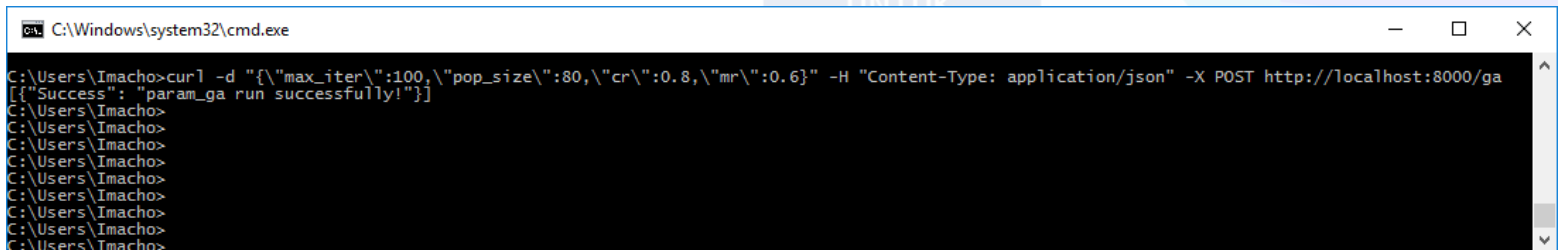
Hex view details:

| Offset | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 0A | 0B | 0C | 0D | 0E | 0F | |
|----------|----|----|----|----|----|----|------------------|----|----|----|----|----|----|----|----|----|--|
| 0000EFAC | 14 | 01 | 04 | 00 | 29 | 02 | 54 | 65 | 73 | 6C | 61 | 20 | 4D | 6F | 64 | 65 | |
| 0000EFBC | 6C | 20 | 53 | 53 | 00 | 9B | ...).Tesla Model | | | | | | | | | | |
| | | | | | | | 1 SS.. | | | | | | | | | | |

22 Bytes Ln 1 Col 1

Python Django (Web)

- Tahapan Koding RESRful di Python Django (Web):
 - Coba ketik “`curl -d '{"max_iter":100,"pop_size":80,"cr":0.8,"mr":0.6}' -H "Content-Type: application/json" -X POST http://localhost:8000/ga`” untuk kemudian dilakukan proses algoritma genetika



```

C:\Windows\system32\cmd.exe
C:\Users\Imacho>curl -d '{"max_iter":100,"pop_size":80,"cr":0.8,"mr":0.6}' -H "Content-Type: application/json" -X POST http://localhost:8000/ga
[{"Success": "param_ga run successfully!"}]
C:\Users\Imacho>
C:\Users\Imacho>
C:\Users\Imacho>
C:\Users\Imacho>
C:\Users\Imacho>
C:\Users\Imacho>
C:\Users\Imacho>
C:\Users\Imacho>
C:\Users\Imacho>

```



Python Django (Web)

- Tampilan Koding RESRful di Python Django (Web):

The screenshot shows a web application interface for 'Masukkan Parameter' (Alg. Genetika). The interface includes a sidebar with user information (Alexander Pierce, Online) and a menu with options like 'Forms', 'Input Parameter', 'Advanced Elements', 'Editors', 'Hasil', and 'Keterangan'. The main content area contains four input fields: 'Jumlah Generasi' (10), 'Ukuran Populasi' (7), 'Nilai Cr' (0.1), and 'Nilai Mr' (0.9). A 'Submit' button is located below these fields. To the right, a 'Log Proses Komputasi' section displays the results of a genetic algorithm simulation, including the task parameters, the final result, and a table of chromosome data.

Masukkan Parameter (Alg. Genetika)

Jumlah Generasi: 10

Ukuran Populasi: 7

Nilai Cr: 0.1

Nilai Mr: 0.9

Submit

Log Proses Komputasi

Running ...
task=10,7,0.1,0.9

Hasil Final (dari RESTful Api Python):

Start Solving Case : Max, $y = f(x) = -x^2 + 14x - 13$, $0 \leq x \leq 15$:

Hasil Biner= 1111
StringLen Chromosome= 4
Hasil Biner2Dec (1111)= 15

| Chromosome | x | y=f(x) |
|---------------|----|--------|
| P1: [0 0 1 1] | 3 | 20.0 |
| P2: [0 0 0 0] | 0 | -13.0 |
| P3: [1 1 1 1] | 15 | -28.0 |
| P4: [1 1 0 0] | 12 | 11.0 |

Tambahan Python

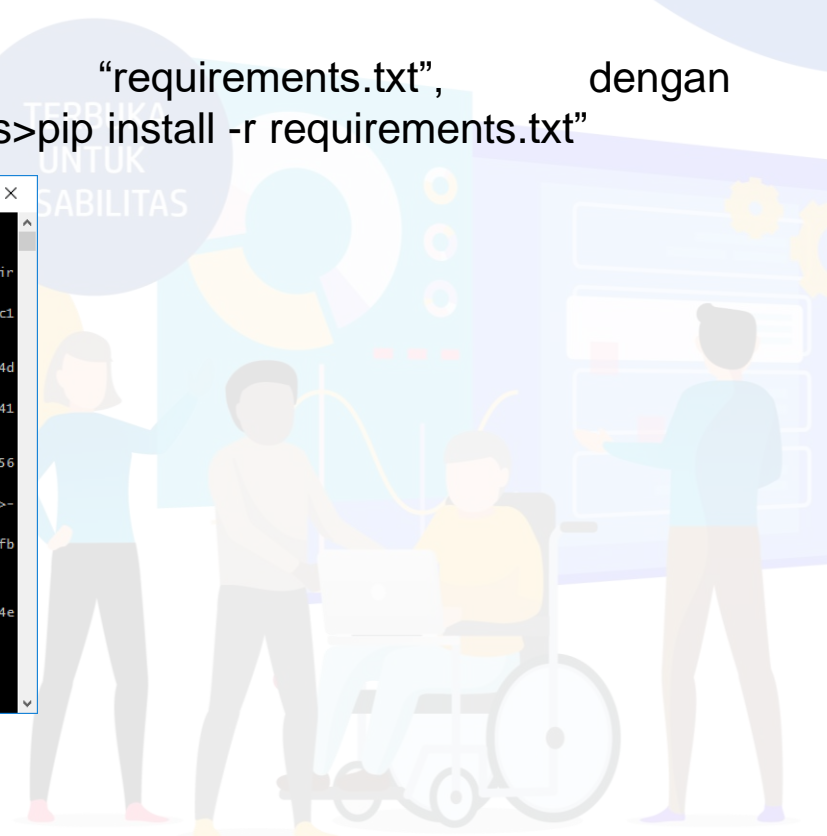
- Tambahan di Python Flask (Web):
 - Cara install dari file “requirements.txt”, dengan
“E:\PyWebProject\NBflask\tryRESTful\tests>pip install -r requirements.txt”

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

E:\PyWebProject\NBflask\tryRESTful\tests>pip install -r requirements.txt
Requirement already satisfied: nose==1.1.2 in e:\installedc\anaconda3\lib\site-packages (from -r requirements.txt (line 1)) (1.3.7)
Collecting mock==0.8 (from -r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/e6/35/f187bdf23be87092bd0f1200d43d23076cee4d0dec109f195173fd3ebc79/mock-2.0.0-py2.py3-none-any.whl (56kB)
    100% |#####| 61kB 38kB/s
Collecting nosexcover (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/f5/74/0dfbedcab931df02d9437820014fd6f4b539b3aa64dbb8e7a362fe20343d/nosexcover-1.0.11-py2.py3-none-any.whl
Collecting blinker (from -r requirements.txt (line 4))
  Downloading https://files.pythonhosted.org/packages/1b/51/e2a9f3b757eb802f61dc1f2b09c8c99f6eb01cf06416c0671253536517b6/blinker-1.4.tar.gz (11kB)
    100% |#####| 112kB 41kB/s
Collecting pbr==0.11 (from mock==0.8->-r requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/b3/5d/c196041ffdf34ba206db6d61d1f893a75e1f3435699ade9bd65e089a3d/pbr-4.0.4-py2.py3-none-any.whl (98kB)
    100% |#####| 102kB 37kB/s
Requirement already satisfied: six>=1.9 in e:\installedc\anaconda3\lib\site-packages (from mock==0.8->-r requirements.txt (line 2)) (1.11.0)
Collecting coverage==3.4 (from nosetest==1.3.7->-r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/4e/df/4d63225d3f39e950cac58b174a3361b7a29c99f51fb8aba0ab4195dcfb71/coverage-4.5.1-cp36-cp36m-win_amd64.whl (181kB)
    100% |#####| 184kB 37kB/s
Building wheels for collected packages: blinker
  Running setup.py bdist_wheel for blinker ... done
  Stored in directory: C:\Users\Imacho\AppData\Local\pip\Cache\wheels\92\1a\00\8690a57883956a301d91cf4e999cc0b258b01e3f548f86e89
Successfully built blinker
Installing collected packages: pbr, mock, coverage, nosetest, blinker
Successfully installed blinker-1.4 coverage-4.5.1 mock-2.0.0 nosetest-1.0.11 pbr-4.0.4

E:\PyWebProject\NBflask\tryRESTful\tests>
  
```



Tambahan Python

- Tambahan di Python Flask (Web):
 - Install “pip install pyfcm”
 - Install “pip install fcm-django”
 - Install “pip install pusher_push_notifications”



Spark Streaming

TERBUKA
UNTUK
DISABILITAS

BREAK
LIMITS



Spark Streaming

- pySparkWordCount streaming:

1. Code `hdfs_wordcount.py` streaming berhasil dijalankan. Link file kode "<https://goo.gl/vY6f4E>"

Karena file kode tersebut streaming, maka akan selalu mencari file baru untuk diproses kembali dengan konsep wordcount setiap waktu (misal per detik)

Pada saat koding dijalankan, untuk melakukan proses wordcount maka masukkan file text sembarang ke alamat `/user/hduser/wordcount/input` pada hdfs, misal file tersebut adalah `input.txt` dan `input2.txt`, dari link berikut "<https://goo.gl/6d7CWQ>"

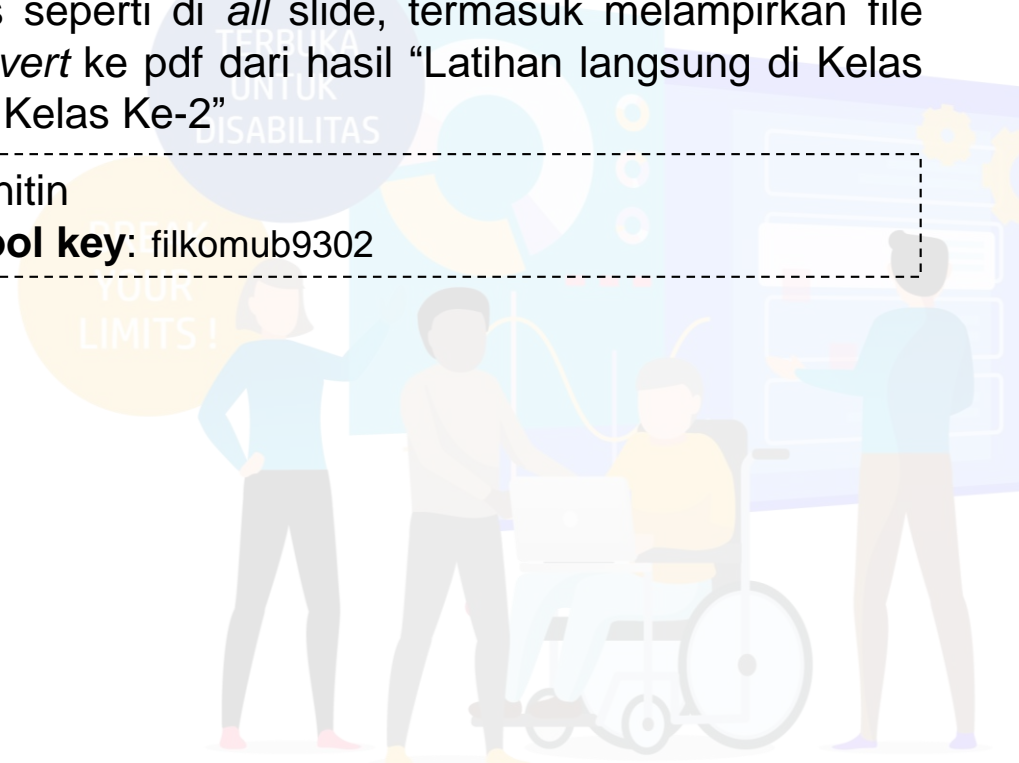
2. Jalankan `pySparkWordCount` streaming ke-2. Link file kode "<https://goo.gl/cnGuHo>"

Tugas Individu

1. Buatlah rangkuman materi di atas dengan cara berikut:

- Lakukan ulang materi di atas seperti di *all* slide, termasuk melampirkan file *.doc/docx yang sudah di-*convert* ke pdf dari hasil “Latihan langsung di Kelas Ke-1 dan Latihan langsung di Kelas Ke-2”

> Register ke turnitin cek plagiasi diturnitin
> Masukkan **id class**: 21563495 & **enroll key**: filkomub9302





DIGITAL TALENT SCHOLARSHIP 2019

Big Data Analytics



Terimakasih

Oleh: Imam Cholissodin | imamcs@ub.ac.id, Putra Pandu Adikara, Sufia Adha Putri

Asisten: Guedho, Sukma, Anshori, Aang dan Gusti

Fakultas Ilmu Komputer (Filkom) Universitas Brawijaya (UB)