Panduan Tugas Besar

Deskripsi Tugas

Buatlah sebuah OLAP *dashboard* untuk analisis kasus COVID. Anda diberi kebebasan untuk menentukan *objective* yang ingin dicapai dari *dashboard* tersebut. Untuk itu, silakan pilih dataset mana saja yang akan Anda gunakan dari tautan-tautan yang diberikan. Tingkat kerumitan akan memengaruhi nilai Anda.

Tahap Pengerjaan

Pembangunan OLAP *dashboard* untuk *big data* umumnya dilakukan secara *real time* (data otomatis diperbarui secara berkala). Untuk mempermudah pekerjaan Anda, fitur otomatisasi tersebut dihilangkan. Untuk tugas besar ini, berikut detail tahapan* yang wajib Anda lakukan:

1. *Extract* & *transform*: mengambil dataset dari URL yang diberikan, membersihkan data (jika perlu), kemudian menyimpannya ke dalam HDFS (harus menggunakan Python).
2. *Load* data ke dalam Hive. Buatlah database dan tabel di Hive sesuai kebutuhan Anda.
3. *Dimensional modeling*. Rancanglah *fact* dan *dimension table* yang Anda butuhkan, kemudian rancang dan buat pula relasional DB-nya.
4. *Transform*: Agregasikan dataset yang sudah ditaruh ke dalam tabel-tabel Hive sesuai dengan rancangan *dimensional modeling* Anda.
5. *Export* seluruh data untuk *fact table* dan *dimensional table* dari Hive ke CSV.
6. *Load / staging 2*: masukkan data-data *fact* dan *dimensional table* dari poin 5 ke dalam relasional DB yang sudah Anda buat di poin 3. Relasional DB tersebut akan berfungsi sebagai *data warehouse* / *data mart* Anda.
7. Buatlah OLAP *dashboard* untuk analisis COVID (*web-based*). Anda cukup membuat 1 halaman dashboard saja.

*Tahapan yang Anda lakukan adalah versi sangat sederhana, karena fokus tugas besar ini adalah pemanfaatan Hive dalam ETL.

Deliverables

Berikut adalah hal-hal yang perlu Anda kumpulkan.

1. Laporan. Isi minimal dari laporan dijelaskan di dalam dokumen ini, bagian “Konten Laporan”.

2. Kode program. Seluruh kode program yang Anda buat. Untuk perintah-perintah Hive cli, tuliskan di file txt seperti saat kuis dan ujian.

Seluruh *deliverables* dikumpulkan paling lambat 1 hari sebelum pertemuan Minggu 13. Di Minggu 13 Anda perlu mempresentasikan hasil pekerjaan Anda selama max 20 menit. Anda cukup menceritakan dengan singkat *objective* pekerjaan dan dimensional model yang Anda buat, lalu tunjukkan OLAP *dashboard* Anda.

Penilaian

Berikut adalah komponen-komponen yang akan dinilai beserta poinnya.

1. ETL – 40
2. Dimensional model & *data warehouse* / *data mart* – 15
3. Dashboard – 25
4. Laporan dan kode program – 20

Konten Laporan

Laporan yang Anda buat minimal berisi hal-hal berikut.

1. Deskripsi *objective* yang akan Anda capai.
2. *Dimensional model* dan rancangan RDB untuk *data warehouse* / *data mart*.
3. *Task* ETL yang Anda lakukan. Jabarkan dengan lengkap dan detail setiap tahap yang Anda lakukan.
    - Dataset yang digunakan
    - *Cleaning data* (jika ada)
    - Transformasi data (agregat, dll.)
    - dsb.
4. *Screenshot* OLAP *dashboard* beserta deskripsi / penjelasan fungsi dari setiap komponen dalam *dashboard* Anda (apa yang dapat dianalisis, dsb.).

URL Dataset

Silakan pilih dataset yang Anda perlukan.

Vaccine Summary

https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations.csv

Vaccine by Location and Manufacturer

https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-manufacturer.csv

Vaccine by Age Group

https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-age-group.csv

Mobility

https://www.gstatic.com/covid19/mobility/Global_Mobility_Report.csv

Case

https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv

Variant

https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/variants/covid-variants.csv

Panduan *Extract & Load - Python*

Untuk menarik data dari URL (yang mengarah langsung ke data), Anda dapat menggunakan pandas. Sebagai contoh, jika URL yang Anda miliki mengarah ke sebuah file CSV, Anda dapat menggunakan

```
dataframe = pandas.read_csv('<URL to CSV>')
```

Pada contoh di atas, data yang dibaca dari URL akan tersimpan dalam variabel *dataframe* dengan struktur *pandas data frame*.

Untuk menyimpan *pandas data frame* sebagai file CSV di HDFS, Anda perlu membuat koneksi terlebih dahulu dari *python* ke HDFS. Untuk membuat koneksi tersebut, Anda dapat menggunakan objek InsecureClient dari *library hdfs*.

```
from hdfs import InsecureClient
client_hdfs = InsecureClient('http://localhost:9870')
```

`'http://localhost:9870'` pada contoh di atas adalah *host* dan *port default* untuk HDFS di lokal Anda masing-masing. Jika Anda menggunakannya di perangkat lain (misalkan di lab), gantilah dengan *host* dan *port* yang sesuai. Jika perlu *credential*, Anda dapat menambahkan parameter *user* saat membuat objek InsecureClient.

```
client_hdfs = InsecureClient('http://<host>:<port>', user = '<username>')
```

Setelah terhubung dengan HDFS, Anda dapat menggunakan fungsi *write* sebagai berikut untuk menuliskan isi dari *pandas data frame* ke dalam file CSV di HDFS.

```
with client_hdfs.write('<hdfs_path>/<file_name>', encoding='utf-8') as writer:
    dataframe.to_csv(writer, index = False)
```

*`index = False` diperlukan untuk menghapus *default index* dari struktur data frame. Cobalah hapus parameter *index*, kemudian lihat perbedaan kolom CSV yang dihasilkan.

= SELAMAT MENGERJAKAN