

HandsOn W05 – MapReduce untuk Data Tabel

Diberikan dataset yang sama, “purchases.txt”, yang digunakan di *Example 02* pada slide, buatlah program mapreduce untuk masing-masing milestone di bawah ini. Untuk menjalankan mapreduce di Hadoop, file “purchases.txt” tersebut harus sudah ditempatkan di suatu folder di HDFS. Sebelumnya, pastikan Hadoop sudah berjalan di VM yang digunakan (seperti yang telah dilakukan di HandsOn W05 sebelumnya).

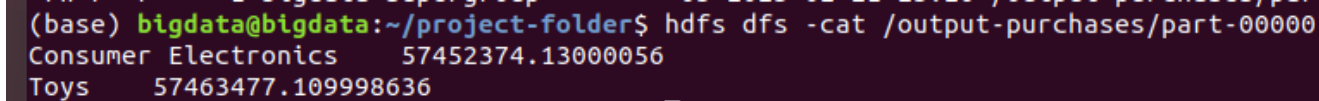
A. Milestone 1

1. Tampilkan total nilai penjualan untuk produk: (i) “Toys” dan (ii) “Consumer Electronics”. Sebagai catatan, nama produk dapat bermacam-macam, selama mengandung salah satu dari kedua string, (i) dan (ii), tersebut. Contoh: “Buffalo Toys”. Output dari milestone ini adalah sebagai berikut.

Consumer Electronics	57452374.13000056
Toys	57463477.109998636

2. Tampilkan hasil MapReduce-nya dalam terminal menggunakan perintah `hdfs dfs -cat /folder_output_kamu/file_output`, dan pastekan screenshotnya di bawah.

<pastekan screenshot di sini> ¹



```
(base) bigdata@bigdata:~/project-folder$ hdfs dfs -cat /output-purchases/part-00000
Consumer Electronics 57452374.13000056
Toys 57463477.109998636
```

Catatan: semua file python mapper dan reducer yang digunakan selama HandsOn ini, dibutuhkan untuk disubmit. Baca detail format pengumpulannya di bagian paling bawah dari dokumen ini.

B. Milestone 2:

1. Tampilkan nilai penjualan tertinggi beserta item produknya² untuk masing-masing toko yang berada di kota: **Miami**, **San Francisco** dan **Atlanta**. Output dari milestone ini adalah sebagai berikut.

Atlanta	499.96	Pet Supplies
Miami	499.98	Video Games
San Francisco	499.97	Men's Clothing

2. Tampilkan hasil MapReducenya dalam terminal menggunakan perintah `hdfs dfs -cat /folder_output_kamu/file_output`, dan pastekan screenshotnya di bawah.

<pastekan screenshot di sini>

¹ *Screenshot* setidaknya memuat hasil dari output MapReduce. Jika hasilnya menyita banyak *space*, ambil *screenshot* secukupnya bagian teratas dari output tersebut

² Bisa jadi penjualan tertinggi nilainya tidak unik dan terdapat pada beberapa produk. Pada kasus yang demikian, kamu hanya perlu mencantumkan salah satu produknya saja

```
(base) bigdata@bigdata:~/project-folder/milestone-2$ hdfs dfs -cat /output-milestone-2/part-00000
Atlanta 499.96 Pet Supplies
Miami 499.98 Video Games
San Francisco 499.97 Men's Clothing
```

C. Milestone 3:

1. Tampilkan banyaknya penjualan yang terjadi di rentang jam 09:01-10:00 dan jam 10:01-11:00. Output dari milestone ini adalah sebagai berikut.

09:01-10:00	459775
10:01-11:00	459825

3. Tampilkan hasil MapReducenya dalam terminal menggunakan perintah `hdfs dfs -cat /folder_output_kamu/file_output`, dan pastekan screenshotnya di bawah.

<pastekan screenshot di sini>

```
(base) bigdata@bigdata:~/project-folder/milestone-3$ hdfs dfs -cat /output-milestone-3/part-00000
09:01-10:00 459775
10:01-11:00 459825
```

Pesan antara:

Implementasi MapReduce dengan membuat file kode secara *custom* untuk mapper dan reducer yang dilakukan di atas memberikan keleluasaan programmer untuk mengembangkan programnya. Akan tetapi, hal tersebut memang diperlukan usaha yang relatif besar untuk membawa permasalahan-permasalahan yang diberikan ke paradigma “map” dan “reduce”. Usaha ini sebanding dengan keuntungan yang bisa kita dapatkan, yaitu mampu mendistribusikan komputasi ke mesin-mesin dalam kluster.

Di dalam ekosistem Hadoop, tersedia sebuah *tool* yang mengubah *SQL-like query* ke komputasi MapReduce yang kemudian dapat didistribusikan ke dalam kluster, yaitu Apache Hive. Dengan menggunakan Apache Hive, seorang programmer dapat mengolah data tabel yang tersimpan (terdistribusi) di kluster layaknya melakukan *query* menggunakan SQL. Sekali lagi, query tersebut kemudian akan dikonversikan ke MapReduce dan akan memproses datanya secara terdistribusi di dalam kluster.

D. Milestone 4

1. Masuk ke Hive dengan cara seperti yang telah dilakukan di HandsOn W04
2. Buatlah tabel “purchases” dari data “purchases.txt” yang telah disimpan di HDFS. Sebagai referensi, untuk membuat tabel “mahasiswa” dari file (dengan tiga kolom, terpisah dengan koma) yang berada di folder HDFS “/mahasiswa”, dapat dilakukan dengan kode berikut.

```
CREATE TABLE IF NOT EXISTS mahasiswa(ID int, nama string, ipk float)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/mahasiswa/';
```

3. Setelah tabel “purchases” terbuat, tes dengan query `select * from purchases limit 10;`
4. Ambil screenshot (hasil query-nya) dan pastekan di bawah ini.

<pastekan screenshot di sini>

```
-hive> select * from purchases limit 10;
-OK
-2012-01-01      09:00      San Jose      Men's Clothing      214.05      Amex
-2012-01-01      09:00      Fort Worth      Women's Clothing      153.57      Visa
-2012-01-01      09:00      San Diego      Music      66.08      Cash
-2012-01-01      09:00      Pittsburgh      Pet Supplies      493.51      Discover
-2012-01-01      09:00      Omaha      Children's Clothing      235.63      MasterCard
-2012-01-01      09:00      Stockton      Men's Clothing      247.18      MasterCard
-2012-01-01      09:00      Austin      Cameras      379.6      Visa
-2012-01-01      09:00      New York      Consumer Electronics      296.8      Cash
-2012-01-01      09:00      Corpus Christi      Toys      25.38      Discover
-2012-01-01      09:00      Fort Worth      Toys      213.88      Visa
Time taken: 4.087 seconds, Fetched: 10 row(s)
```

E. Milestone 5

1. Lakukan Milestone 1, akan tetapi menggunakan query Hive dari tabel “purchases” yang telah dibuat.³
2. Ambil screenshot (bagian ekspresi SQL dan hasil query-nya), dan pastekan di bawah ini. Hasil Milestone 1 dan 5 seharusnya memberikan keluaran yang sama⁴.

<pastekan screenshot di sini>

Ekspresi SQL:

```
hive> select item, sum(harga) from purchases where item in ('Toys', 'Consumer Electronics') group by item;
```

Hasil Query:

```
hive> select item, sum(harga) from purchases where item in ('Toys', 'Consumer Electronics') group by item;
Query ID = bigdata_20230222020220_643a909d-9c3c-4fb5-b634-af0161033047
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1676481430158_0009, Tracking URL = http://bigdata:8088/proxy/application_1676481430158_0009
Kill Command = /home/bigdata/hadoop-3.2.2/bin/mapred job -kill job_1676481430158_0009
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-02-22 02:02:36,993 Stage-1 map = 0%, reduce = 0%
2023-02-22 02:02:50,124 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 5.43 sec
2023-02-22 02:02:59,788 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 7.28 sec
MapReduce Total cumulative CPU time: 7 seconds 280 msec
Ended Job = job_1676481430158_0009
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 7.28 sec HDFS Read: 211331520 HDFS Write: 177 SUCCESS
Total MapReduce CPU Time Spent: 7 seconds 280 msec
OK
Consumer Electronics      5.745237412163785E7
Toys      5.746347711329021E7
Time taken: 40.444 seconds, Fetched: 2 row(s)
```

³ Bagi VM dengan size RAM kecil, kemungkinan proses akan terhenti di tengah. Jika tidak memungkinkan untuk menambahkan size RAM di VM, ambil screenshot “ekspresi SQL yang kamu buat” dan “pesan errornya”.

⁴ Hiraukan perbedaan minor string “lowercase” dan “Capital Each Word”.

F. Milestone 6

1. Lakukan Milestone 2, akan tetapi menggunakan query Hive dari tabel “purchases” yang telah dibuat.
2. Ambil screenshot (bagian ekspresi SQL dan hasil query-nya), dan pastekan di bawah ini. Hasil Milestone 2 dan 6 seharusnya memberikan keluaran yang sama.

<pastekan screenshot di sini>

Ekspresi SQL:

```
hive> with t1 as (select max(harga) as MX, kota from purchases where kota = 'Atlanta' group by kota),
> t2 as (select max(harga) as MX, kota from purchases where kota = 'Miami' group by kota),
> t3 as (select max(harga) as MX, kota from purchases where kota = 'San Francisco' group by kota)
>
> (select purchases.kota, harga, item from purchases join t1 on purchases.kota = t1.kota and purchases.harga = t1.MX limit 1)
> union
> (select purchases.kota, harga, item from purchases join t2 on purchases.kota = t2.kota and purchases.harga = t2.MX limit 1)
> union
> (select purchases.kota, harga, item from purchases join t3 on purchases.kota = t3.kota and purchases.harga = t3.MX limit 1);
```

Hasil Query:

```
OK
Atlanta 499.96 Pet Supplies
Miami 499.98 Video Games
San Francisco 499.97 Men's Clothing
Time taken: 408.431 seconds, Fetched: 3 row(s)
```

G. Milestone 7

1. Lakukan Milestone 3, akan tetapi menggunakan query Hive dari tabel “purchases” yang telah dibuat.
2. Ambil screenshot (bagian ekspresi SQL dan hasil query-nya), dan pastekan di bawah ini. Hasil Milestone 3 dan 7 seharusnya memberikan keluaran yang sama.

<pastekan screenshot di sini>

Ekspresi SQL:

```
hive> with t1 as (select '09:01-10:00' as itv, count(*) as cnt from purchases where jam >= '09:01' and jam <= '10:00'),
> t2 as (select '10:01-11:00' as itv, count(*) as cnt from purchases where jam >= '10:01' and jam <= '11:00')
>
> (select * from t1)
> union
> (select * from t2);
```

Hasil Query:

```
Total MapReduce CPU Time Spent: 15 seconds 730 ms
OK
09:01-10:00 459775
10:01-11:00 459825
Time taken: 103.629 seconds, Fetched: 2 row(s)
```

H. Milestone 8

1. Jika pada Milestone 5, 6 dan 7 hasil yang didapatkan tidak dapat menyamai⁵ dari Milestone 1, 2 dan 3, berikan analisis kamu. Jika alasannya adalah terkait keterbatasan SQL-like query, berikan ide/solusinya agar hasil dari Milestone 5, 6 dan 7 secara berturut-turut sama dengan Milestone 1, 2 dan 3.

Hasilnya sama, sehingga tidak perlu ada penjelasan dan analisis lebih lanjut. Kecuali untuk Milestone 1 dan Milestone 5, walaupun hasilnya sama, namun presentasi hasilnya berbeda (penulisan notasinya berbeda, yang satu menggunakan notasi desimal, yang satunya menggunakan notasi saintifik dengan penggunaan huruf E untuk menunjukkan eksponen dari 10).

Setelah semua screenshot di-pastekan di masing-masing milestone, upload file zip dengan nama: “W05_NIM_NamaLengkap.zip” ke form submission/assignment di **edunex** yang telah disediakan.

Adapun isi dari file zipnya adalah:

1. File pdf dari dokumen ini, dengan nama: “W05_NIM_NamaLengkap.pdf”
2. File mapper dan reducer dari Milestone 1-3, dengan format nama “mapper_milestone1.py” dan “reducer_milestone1.py” untuk Milestone 1, begitu seterusnya hingga Milestone 3.

--- done ---

⁵ Hiraukan perbedaan minor string “lowercase” dan “Capital Each Word”.