HandsOn W05 – MapReduce untuk Data Tabel

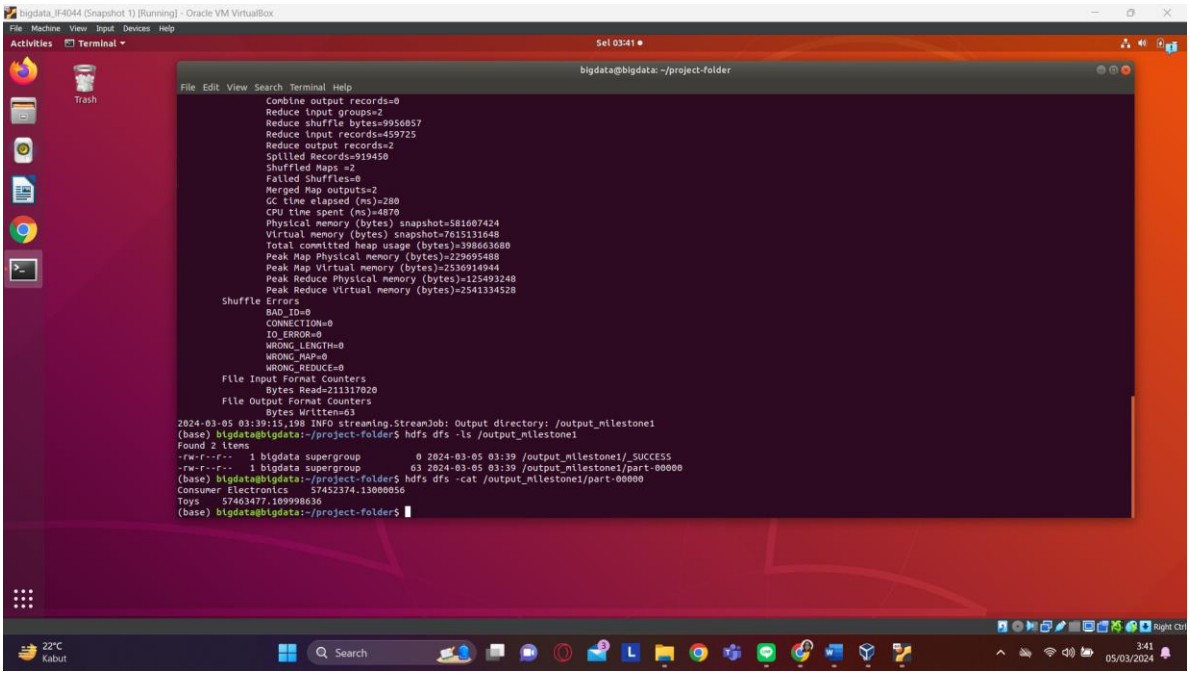
Diberikan dataset yang sama, “purchases.txt”, yang digunakan di ***Example 02*** pada slide, buatlah program map reduce untuk masing-masing milestone di bawah ini. Untuk menjalankan map reduce di Hadoop, file “purchases.txt” tersebut harus sudah ditempatkan di suatu folder di HDFS. Sebelumnya, pastikan 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 <jumlah_nilai_penjualan> Toys <jumlah_nilai_penjualan>

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



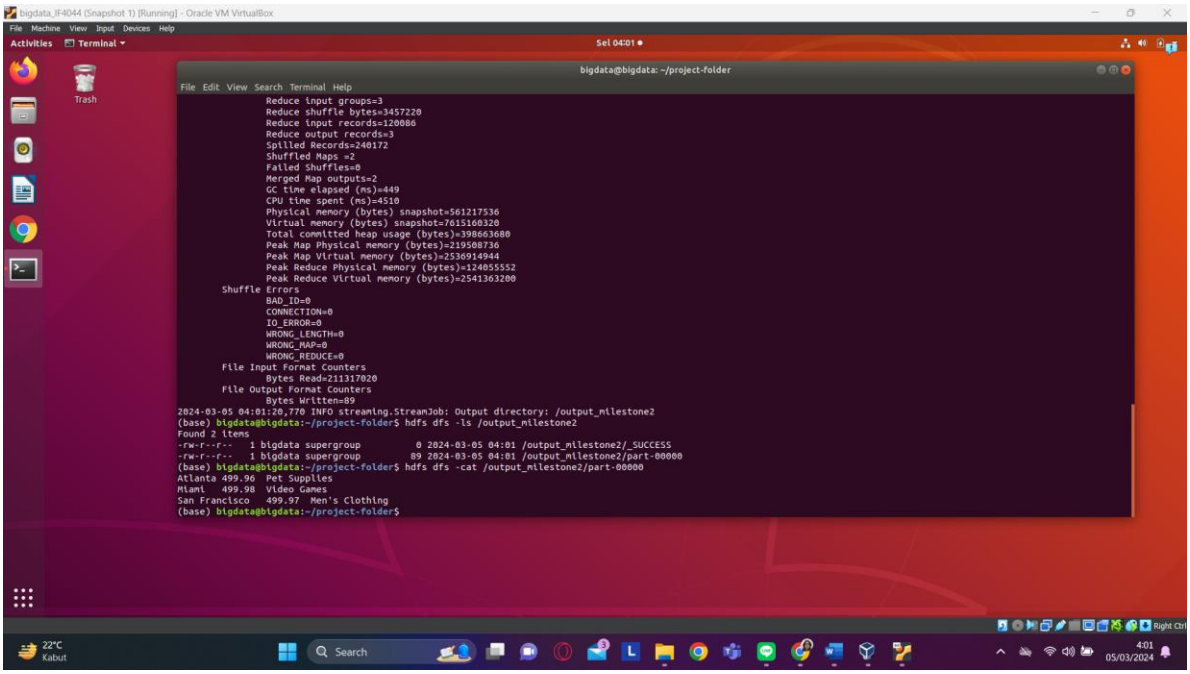
Catatan: semua file python mapper dan reducer yang digunakan selama HandsOn ini, dibutuhkan untuk disubmit. Baca detail format pengumpulannya di bagian paling bawah dari dokumen ini.

B. Milestone 2:

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

Atlanta	<nilai_penjualan_tertinggi>	<item_produk>
Miami	<nilai_penjualan_tertinggi>	<item_produk>
San Francisco	<nilai_penjualan_tertinggi>	<item_produk>

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



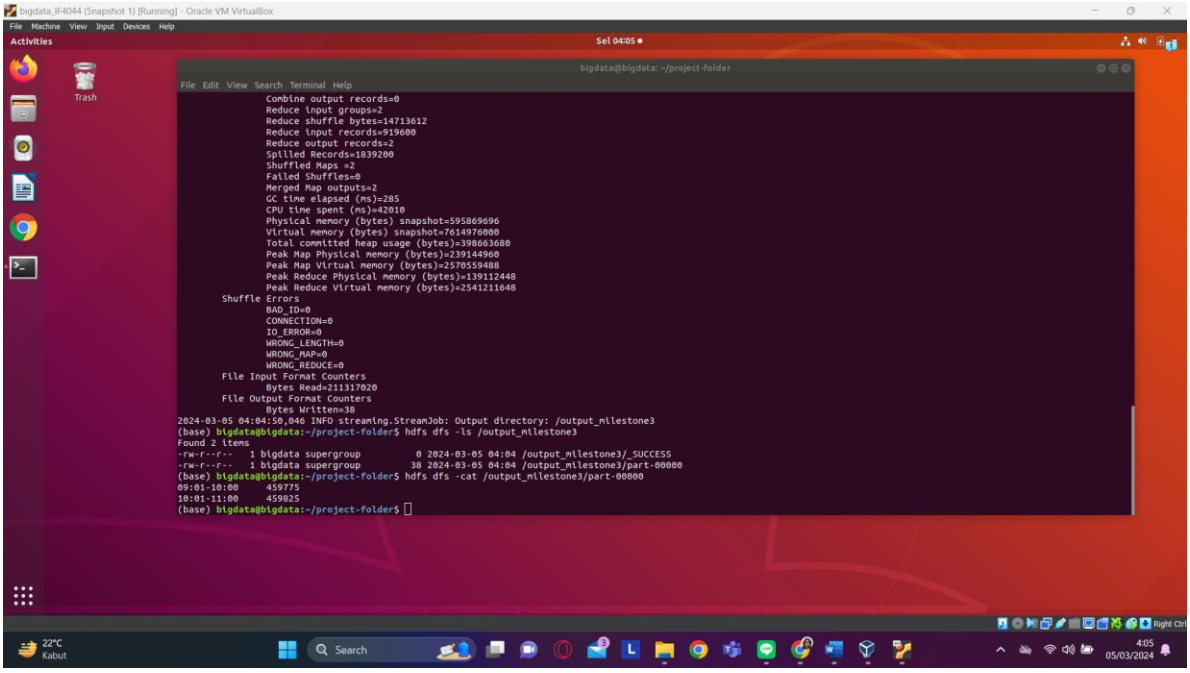
C. Milestone 3:

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	<banyaknya_penjualan_yang_terjadi>
10:01-11:00	<banyaknya_penjualan_yang_terjadi>

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

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



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-permasalah 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.

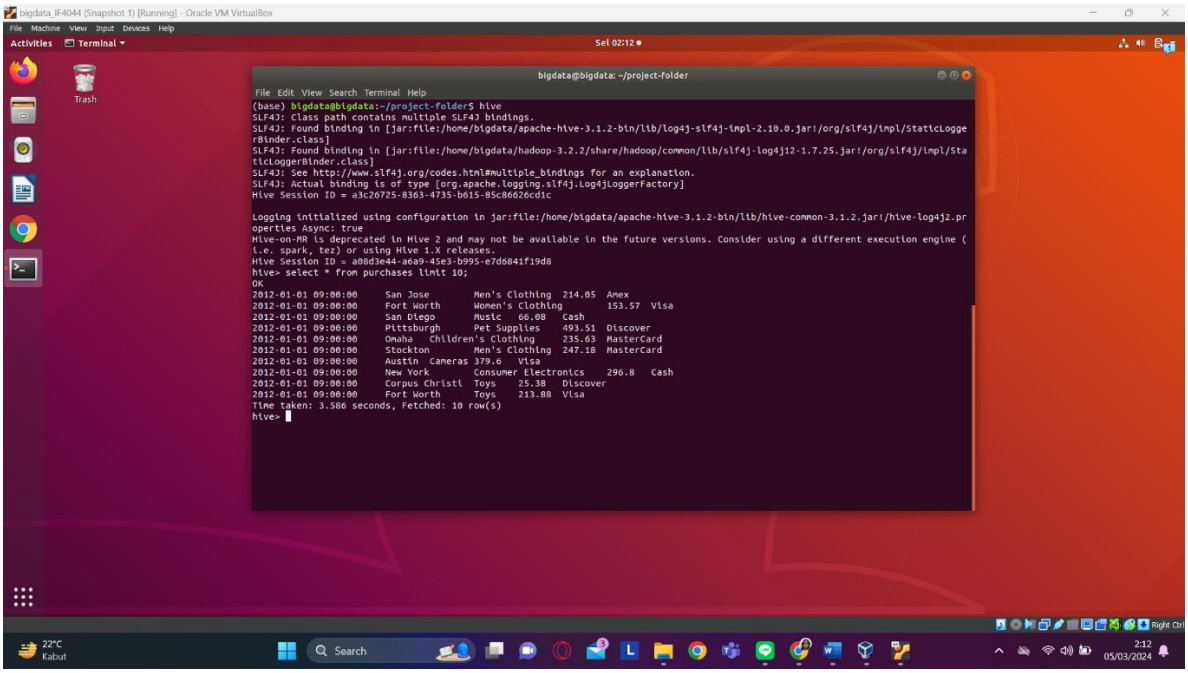
Di dalam ekosistem Hadoop, tersedia sebuah *tool* yang mengubah *SQL-like query* ke komputasi MapReduce yang kemudian dapat didistribusikan ke dalam klaster, yaitu Apache Hive. Dengan menggunakan Apache Hive, seorang programmer dapat mengolah data tabel yang tersimpan (terdistribusi) di klaster layaknya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

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.

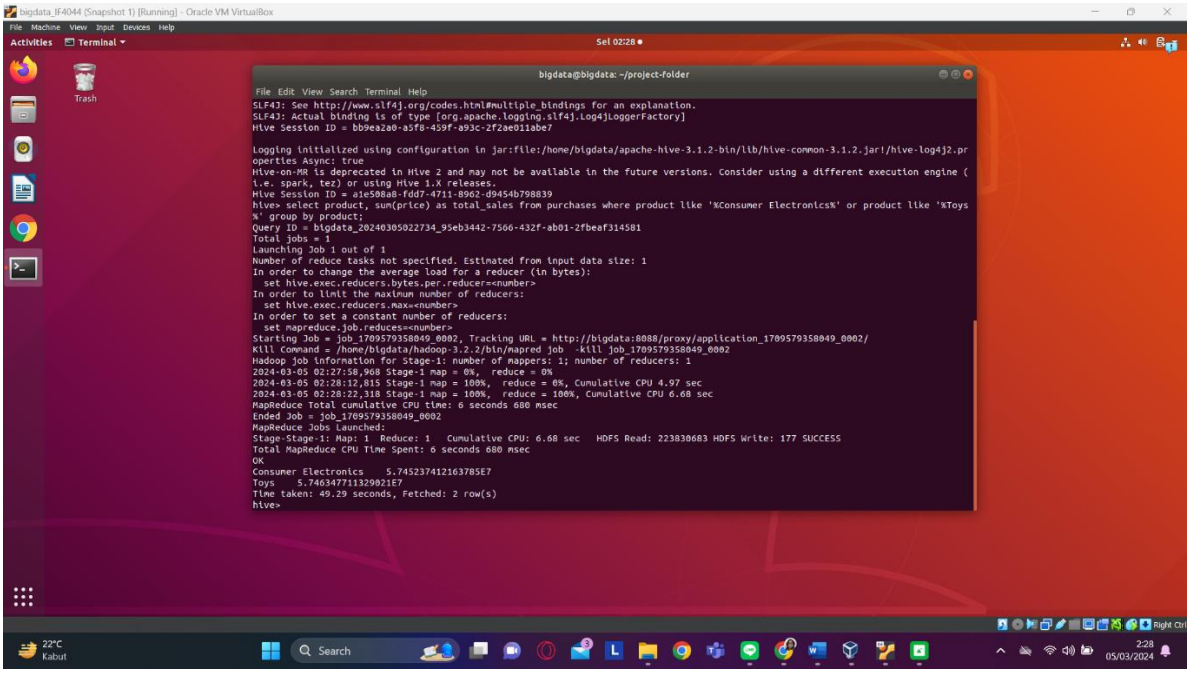
```
CREATE TABLE IF NOT EXISTS mahasiswa(ID int, nama string, ipk float)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION '/mahasiswa/';
```

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



E. Milestone 5

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

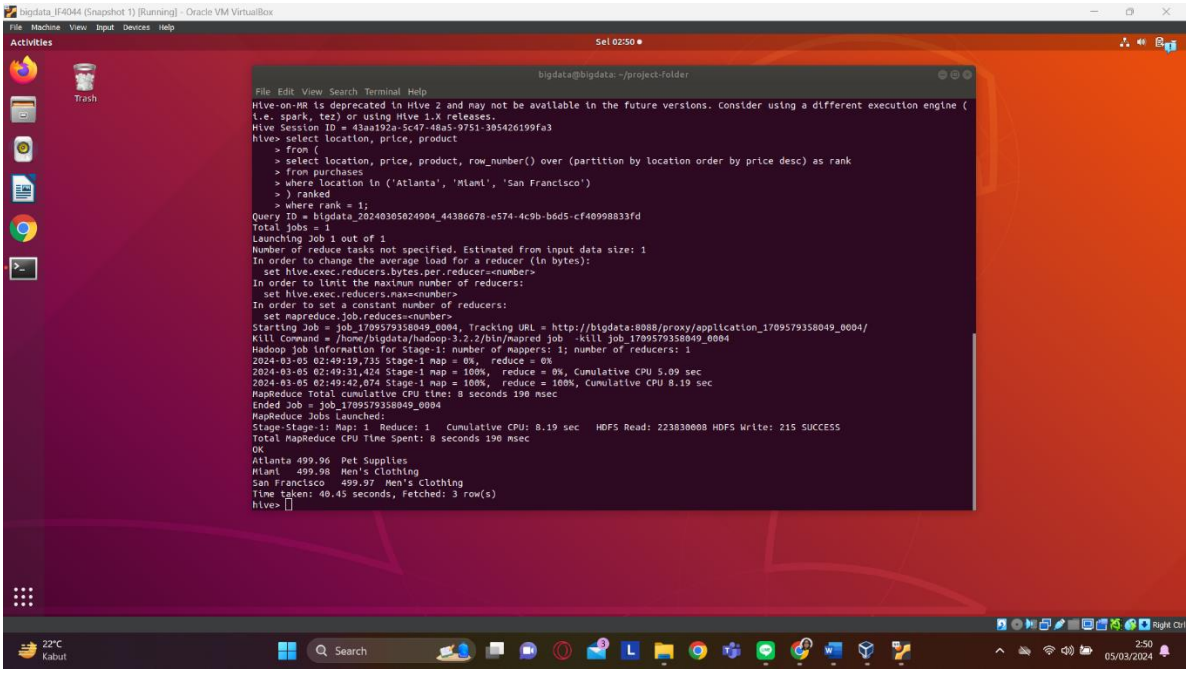


[2] Bagi VM dengan size RAM kecil, kemungkinan proses akan terhenti di tengah. Jika tidak memungkinan untuk menambahkan size RAM di VM, ambil screenshot “ekspresi SQL yang kamu buat” dan “pesan errornya”.

[3] Hiraukan perbedaan minor string “lowercase” dan “Capital Each Word”.

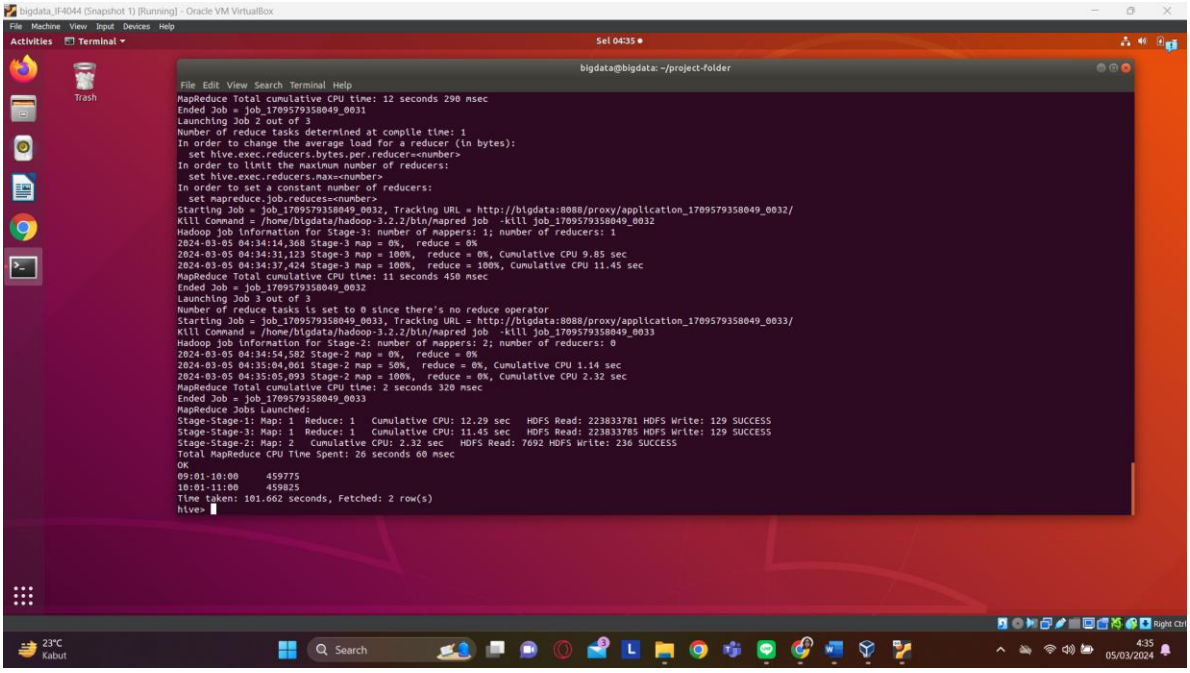
F. Milestone 6

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.



G. Milestone 7

1. Lakukan Milestone 3, akan tetapi menggunakan query Hive dari tabel “puchases” 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.



H. Milestone 8

1. Jika pada Milestone 5, 6 dan 7 hasil yang didapatkan tidak dapat menyamai[4] dari Milestone 1, 2 dan 3, berikan analisa 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.

   Berdasarkan handson yang telah dilakukan, terdapat perbedaan hasil antara milestone 1 dengan milestone 5 dan milestone 2 dengan milestone 6.

   Perbedaan hasil antara milestone 1 dengan milestone 5 terletak pada angka yang terdapat dibelakang koma pada hasil penjumlahan data nilai penjualan produk yang ada. Hal yang dapat dilakukan adalah dengan lebih mendefenisikan jumlah angka dibelakang koma, dengan cara seperti itu maka hasil penjumlahan nilai penjualan akan lebih serupa.

   Perbedaan hasil antara milestone 2 dengan milestone 6 terletak pada data dengan nilai location Miami, pada milestone 2 nilai product nya adalah Video Games sedangkan pada milestone 6 nilai product nya adalah Men's Clothing, meskipun nilai penjualan pada dua product tersebut sama. Hal tersebut dapat terjadi karena pada reducer yang telah dibuat pencarian yang dilakukan hanya mencari nilai penjualan maksimum, tetapi tidak mengupdate value dari product apabila terdapat product yang memiliki nilai penjualan serupa. Sehingga ketika record dengan product Video Games muncul lebih awal, product yang akan tercatat adalah Video Games. Namun, ketika implementasi menggunakan hive yang merupakan SQL-like query record yang ada diurutkan berdasarkan nilai penjualan kemudian nama product sehingga product yang tercatat adalah Men's Clothing.

[4] Hiraukan perbedaan minor string "lowercase" dan "Capital Each Word".

Setelah semua screenshot dipastekan 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 dari Milestone 1-3, dengan format nama "mapper_milestone1.py" dan "reducer_milestone1.py" untuk Milestone 1, begitu seterusnya hingga Milestone 3.

--- done ---