

LAPORAN
RENCANA TUGAS MANDIRI (RTM) Ke-4
MATA KULIAH BIG DATA (A)
“QUERY STATISTIK DEKRIPTIF MENGGUNAKAN HIVE DAN
XQUERY”



DISUSUN OLEH:

Reza Putri Angga (22083010006)

DOSEN PENGAMPU:

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

PROGRAM STUDI SAINS DATA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR
2024

STUDI KASUS DAN PEMBAHASAN

Melakukan analisis pada beberapa data, meliputi :

1. Data Noaa
 - A. Menerapkan *query* statistik deskriptif (maksimum, minimum, rata-rata, varian, standar deviasi, dan persentil).
 - B. Analisis kenaikan suhu di antara dua tahun
 - C. Korelasi antara dua variabel (korelasi positif, korelasi negatif, tidak ada korelasi)
 - D. Deteksi nilai ekstrem pada suhu di tiap-tiap tahun
2. Data Log Trading Saham dan Data Harga Sembako
 - A. Menerapkan XQuery Transformation.

Untuk melakukan pembahasan dari studi kasus tersebut, akan dipecah menjadi beberapa bagian menurut dataset yang dipergunakan dan akan dijelaskan lebih lanjut sebagai berikut.

A. Data Noaa

Data Noaa (*National Oceanic and Atmospheric Administration*) terdiri dari kolom tahun, suhu, dan kualitas. Pada penugasan ini dilakukan penggunaan data Noaa dari tahun 1901 sampai dengan tahun 1932. Dimana, sebelum dilakukan analisis lebih lanjut perlu dilakukan proses pengunduhan data Noaa menggunakan *link google drive* yang berisi data tersebut menggunakan *mozilla* di oracle big data lite.

Kemudian, sebelum melakukan analisis pada data Noaa yang tersedia, dilakukan beberapa proses persiapan pada data tersebut, di antaranya yakni.

```
oracle@bigdatalite:/opt
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ cd /opt
[oracle@bigdatalite opt]$ sudo unzip noaa.zip

oracle@bigdatalite:~
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ sudo mv /home/oracle/Downloads/noaa.zip /opt/
[oracle@bigdatalite ~]$ ls /opt/
cloudera  noaa.zip  oracle  sharedir
rh        VBoxGuestAdditions-4.3.26

oracle@bigdatalite:/opt/noaa
File Edit View Search Terminal Help
[oracle@bigdatalite opt]$ cd noaa
[oracle@bigdatalite noaa]$ ls
1901 1904 1907 1910 1913 1916 1919 1922 1925 1928 1931
1902 1905 1908 1911 1914 1917 1920 1923 1926 1929 1932
1903 1906 1909 1912 1915 1918 1921 1924 1927 1930
```

proses pemindahan, pengecekan, dan penampilan data Noaa ke opt

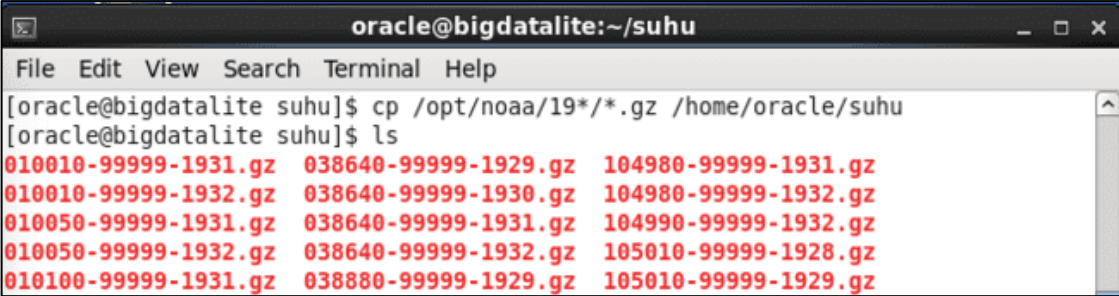
Dilakukan proses pemindahan data Noaa ke dalam *opt* dan penampilan data Noaa yang nantinya akan dilakukan proses *unzip* dan melakukan proses penghapusan pada data Noaa yang rusak.



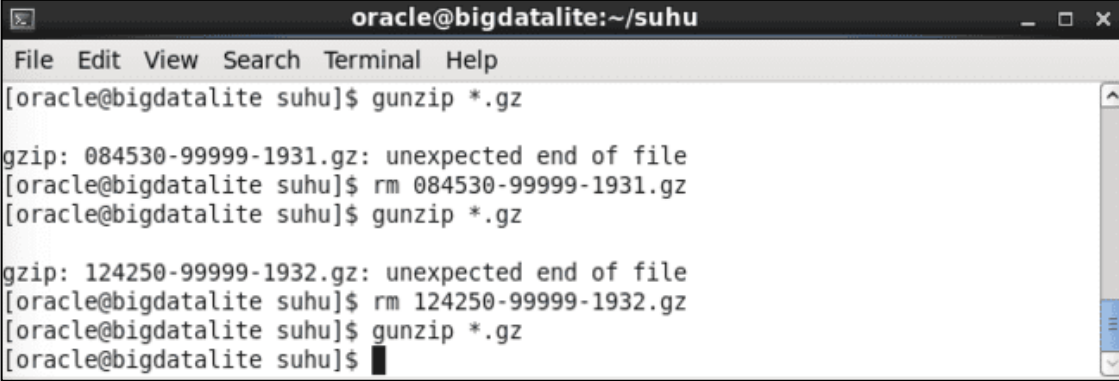
```
oracle@bigdatalite:~/suhu
File Edit View Search Terminal Help
[oracle@bigdatalite noaa]$ cd
[oracle@bigdatalite ~]$ mkdir suhu
[oracle@bigdatalite ~]$ cd suhu
```

proses pembuatan direktori suhu untuk menyimpan data Noaa

Dilakukan proses pembuatan direktori baru bernama suhu untuk menyimpan data Noaa.




```
oracle@bigdatalite:~/suhu
File Edit View Search Terminal Help
[oracle@bigdatalite suhu]$ cp /opt/noaa/19*/*.gz /home/oracle/suhu
[oracle@bigdatalite suhu]$ ls
010010-99999-1931.gz  038640-99999-1929.gz  104980-99999-1931.gz
010010-99999-1932.gz  038640-99999-1930.gz  104980-99999-1932.gz
010050-99999-1931.gz  038640-99999-1931.gz  104990-99999-1932.gz
010050-99999-1932.gz  038640-99999-1932.gz  105010-99999-1928.gz
010100-99999-1931.gz  038880-99999-1929.gz  105010-99999-1929.gz
```



```
oracle@bigdatalite:~/suhu
File Edit View Search Terminal Help
[oracle@bigdatalite suhu]$ gunzip *.gz
gzip: 084530-99999-1931.gz: unexpected end of file
[oracle@bigdatalite suhu]$ rm 084530-99999-1931.gz
[oracle@bigdatalite suhu]$ gunzip *.gz
gzip: 124250-99999-1932.gz: unexpected end of file
[oracle@bigdatalite suhu]$ rm 124250-99999-1932.gz
[oracle@bigdatalite suhu]$ gunzip *.gz
[oracle@bigdatalite suhu]$
```

proses penghapusan data Noaa yang rusak

Dilakukan proses penyalinan data Noaa ke direktori suhu kemudian dilakukan penghapusan sebanyak dua data yang rusak. Kemudian, data Noaa yang dilakukan penghapusan pada data yang rusak tersebut di simpan kedalam *file* baru bernama “suhuall.txt”.



```
oracle@bigdatalite:~
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ cut -c16-19 suhuall.txt > tahun.txt
[oracle@bigdatalite ~]$ cat tahun.txt
```

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
1931  
1931  
1931  
1931  
1931  
[oracle@bigdatalite ~]$
```

proses pengambilan data tahun

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ cut -c90-92 suhuall.txt > suhu.txt  
[oracle@bigdatalite ~]$  
[oracle@bigdatalite ~]$ cat suhu.txt
```

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
135  
124  
124  
124  
119  
[oracle@bigdatalite ~]$
```

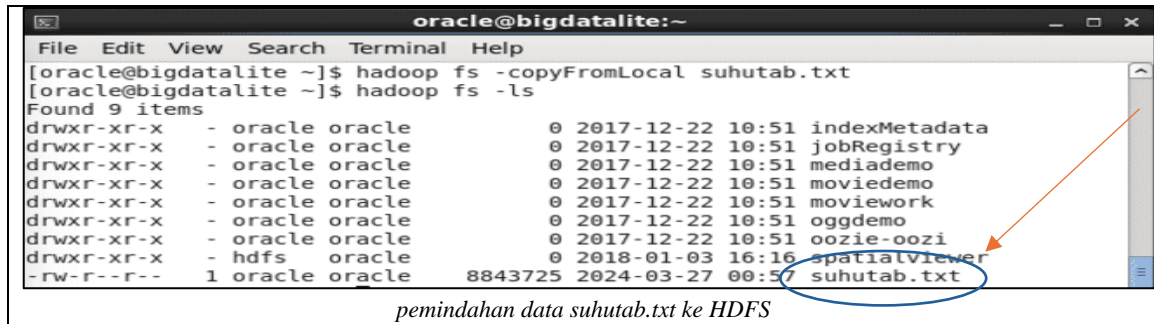
proses pengambilan data suhu

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ cut -c93 suhuall.txt > quality.txt  
[oracle@bigdatalite ~]$ cat quality.txt
```

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
1  
1  
1  
1  
1  
1  
[oracle@bigdatalite ~]$
```

proses pengambilan data quality

Dilakukan proses pembagian data Noaa dengan pengambilan data tahun yang disimpan dalam file “tahun.txt”, data suhu yang disimpan dalam file “suhu.txt”, dan data kualitas yang disimpan dalam file “quality.txt”. Dimana data-data tersebut akan digabungkan kedalam file dengan nama “suhutab.txt” dengan panjang data 803975 dan dilakukan proses pemindahan kedalam hadoop.



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ hadoop fs -copyFromLocal suhutab.txt  
[oracle@bigdatalite ~]$ hadoop fs -ls  
Found 9 items  
drwxr-xr-x - oracle oracle 0 2017-12-22 10:51 indexMetadata  
drwxr-xr-x - oracle oracle 0 2017-12-22 10:51 jobRegistry  
drwxr-xr-x - oracle oracle 0 2017-12-22 10:51 mediademo  
drwxr-xr-x - oracle oracle 0 2017-12-22 10:51 moviedemo  
drwxr-xr-x - oracle oracle 0 2017-12-22 10:51 movieview  
drwxr-xr-x - oracle oracle 0 2017-12-22 10:51 oggdemo  
drwxr-xr-x - oracle oracle 0 2017-12-22 10:51 oozie-oozi  
drwxr-xr-x - hdfs oracle 0 2018-01-03 16:16 spatialviewer  
-rw-r--r-- 1 oracle oracle 8843725 2024-03-27 00:57 suhutab.txt
```

pemindahan data suhutab.txt ke HDFS

Dimana data Noaa yang telah dibersihkan dan diproses disimpan dalam file “suhutab.txt” dalam HDFS akan dilakukan proses analisis dan penerapan *query* statistik menggunakan HIVE. HIVE merupakan *platform* data *warehousing* dan *query* yang memungkinkan pengguna untuk mengakses dan menganalisis data yang disimpan dalam hadoop dengan menggunakan bahasa *query* yang mirip dengan SQL.

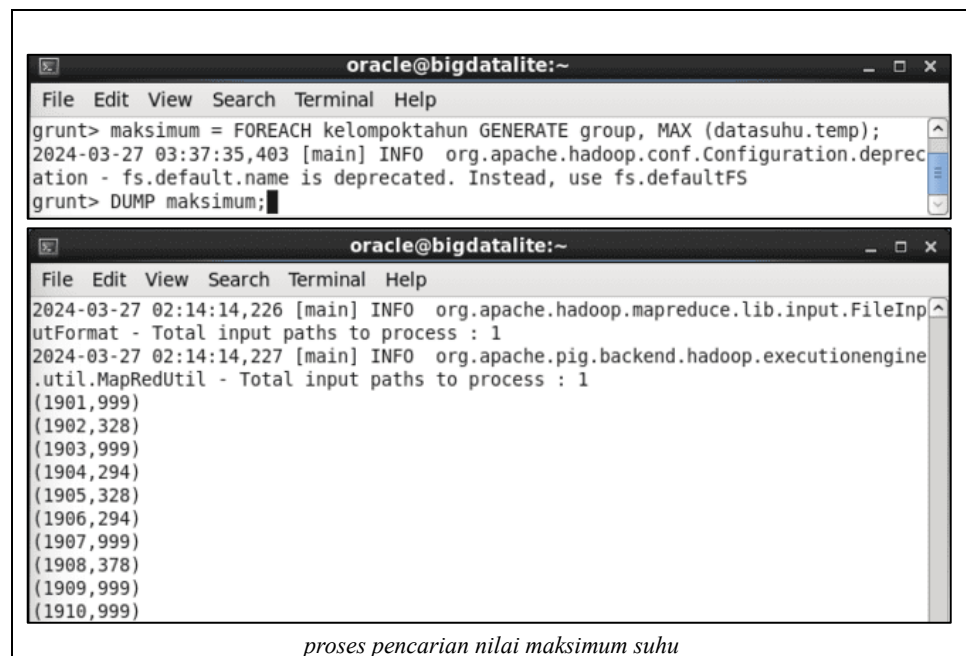
1. Query Statistika Deskriptif Menggunakan PIG Dan Hive

Untuk melakukan *query* statistika deskriptif dapat digunakan PIG dengan pemrograman skrip seperti python dan memungkinkan melakukan penerapan algoritma untuk pemrosesan data dengan cara yang lebih deklaratif dan HIVE dengan menggunakan bahasa *query* seperti SQL.

1.1 Menggunakan PIG

Dengan menggunakan PIG dilakukan analisis statistika deskriptif untuk mencari nilai maksimal, minimal, dan rata-rata. Dengan menjalankan *\$pig* pada terminal dan melakukan inisiasi variabel “datasuhu” yang berisi data dari “suhutab.txt” dan dikelompokkan berdasarkan tahun yang akan diproses lebih lanjut.

1.1.1 Mencari Nilai Maksimum Dan Minimum Dari Suhu



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
grunt> maksimum = FOREACH kelompoktahun GENERATE group, MAX (datasuhu.temp);  
2024-03-27 03:37:35,403 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS  
grunt> DUMP maksimum;
```

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
2024-03-27 02:14:14,226 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 1  
2024-03-27 02:14:14,227 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 1  
(1901,999)  
(1902,328)  
(1903,999)  
(1904,294)  
(1905,328)  
(1906,294)  
(1907,999)  
(1908,378)  
(1909,999)  
(1910,999)
```

proses pencarian nilai maksimum suhu

The first terminal window shows the command: `grunt> minimum = FOREACH kelompoktahun GENERATE group, MIN (datasuhu.temp);` followed by a deprecation warning and `grunt> DUMP minimum;`. The second terminal window shows the output: `(1901,0)`, `(1902,0)`, `(1903,0)`, `(1904,0)`, and `(1905,0)`.

proses pencarian nilai minimum suhu

Dilakukan proses pencarian nilai maksimum dan minimum dari suhu atau *temp* dengan data yang tersimpan dalam variabel *datasuhu*. Dengan menggunakan variabel maksimum untuk mencari nilai maksimum dengan fungsi MAX dan menggunakan variabel minimum untuk mencari nilai minimum dengan fungsi MIN. Diperoleh nilai maksimum suhu disetiap tahun seperti pada tahun 1901 memiliki nilai maksimum suhu sebesar 999 dan nilai minimum suhu di setiap tahun seperti pada tahun 1901 memiliki nilai minimum suhu sebesar 0.

1.1.2 Mencari Nilai Rata-Rata Dari Suhu

The first terminal window shows the command: `grunt> ratarata = FOREACH kelompoktahun GENERATE group, AVG (datasuhu.temp);` followed by `grunt> DUMP ratarata;`. The second terminal window shows the output: `(1901,93.8994668697639)`, `(1902,73.78385376999239)`, `(1903,78.52105584375954)`, `(1904,74.4364934670313)`, `(1905,75.15333028501753)`, `(1906,83.6172816952868)`, and `(1907,89.30807248764415)`.

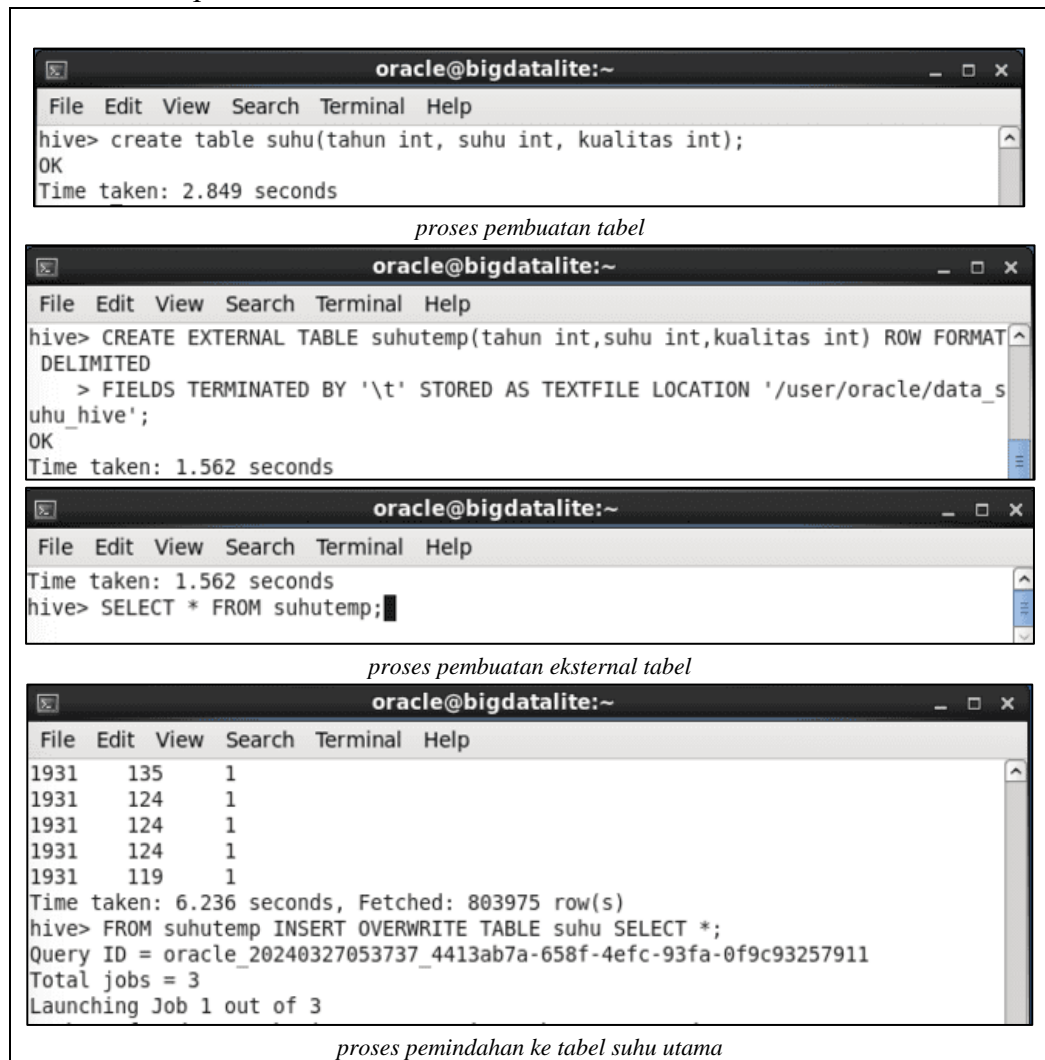
proses pencarian nilai rata-rata dari suhu

Dilakukan proses pencarian nilai rata-rata dari suhu atau *temp* yang tersimpan dalam variabel *datasuhu*. Dengan menggunakan variabel *ratarata* untuk mencari nilai rata-rata suhu disetiap tahun dengan fungsi AVG. Diperoleh nilai rata-rata suhu disetiap tahun, seperti pada tahun 1901 memiliki nilai rata-rata suhu sebesar 93,8995.

1.2 Menggunakan HIVE

Dengan menggunakan HIVE dilakukan analisis statistika deksriptif untuk mencari nilai maksimum dan minimum dari variabel tahun, suhu, dan kualitas. Mencari nilai rata-rata, varian, standar deviasi, dan persentil dari variabel suhu dan kualitas. Dengan menjalankan *\$hive* pada terminal dan melakukan inisiasi pembuatan tabel dengan nama “suhu” sebagai tabel utama dengan kolom tahun, suhu, dan kualitas.

Kemudian, dilakukan proses pembuatan direktori “data_suhu_hive” untuk melakukan *copy* data “suhutab.txt”. Dengan pembuatan eksternal tabel “suhutemp” yang berisi data “suhutab.txt”. Kemudian, “suhutemp” akan dipindah ke tabel utama “suhu” yang telah disimpan dalam hadoop dan untuk melakukan analisis statistika deskriptif.



The image displays four sequential screenshots of a terminal window titled 'oracle@bigdatalite:~'. Each screenshot shows a Hive command and its corresponding output.

Screenshot 1: The command `hive> create table suhu(tahun int, suhu int, kualitas int);` is entered. The output shows 'OK' and 'Time taken: 2.849 seconds'.

Screenshot 2: The command `hive> CREATE EXTERNAL TABLE suhutemp(tahun int,suhu int,kualitas int) ROW FORMAT DELIMITED > FIELDS TERMINATED BY '\t' STORED AS TEXTFILE LOCATION '/user/oracle/data_suhu_hive';` is entered. The output shows 'OK' and 'Time taken: 1.562 seconds'.

Screenshot 3: The command `hive> SELECT * FROM suhutemp;` is entered. The output shows 'Time taken: 1.562 seconds' and a preview of data rows.

Screenshot 4: The command `hive> FROM suhutemp INSERT OVERWRITE TABLE suhu SELECT *;` is entered. The output shows 'Time taken: 6.236 seconds, Fetched: 803975 row(s)', 'Query ID = oracle_20240327053737_4413ab7a-658f-4efc-93fa-0f9c93257911', 'Total jobs = 3', and 'Launching Job 1 out of 3'.

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> create table suhu(tahun int, suhu int, kualitas int);  
OK  
Time taken: 2.849 seconds  
  
proses pembuatan tabel  
  
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> CREATE EXTERNAL TABLE suhutemp(tahun int,suhu int,kualitas int) ROW FORMAT  
DELIMITED  
> FIELDS TERMINATED BY '\t' STORED AS TEXTFILE LOCATION '/user/oracle/data_s  
uhu_hive';  
OK  
Time taken: 1.562 seconds  
  
proses pembuatan eksternal tabel  
  
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
Time taken: 1.562 seconds  
hive> SELECT * FROM suhutemp;  
  
proses pemindahan ke tabel suhu utama  
  
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
1931 135 1  
1931 124 1  
1931 124 1  
1931 124 1  
1931 119 1  
Time taken: 6.236 seconds, Fetched: 803975 row(s)  
hive> FROM suhutemp INSERT OVERWRITE TABLE suhu SELECT *;  
Query ID = oracle_20240327053737_4413ab7a-658f-4efc-93fa-0f9c93257911  
Total jobs = 3  
Launching Job 1 out of 3
```

1.2.1 Nilai Maksimum Dan Minimum Dari Tahun, Suhu, Dan Kualitas

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT MIN(tahun), MAX(tahun) from suhu;  
Query ID = oracle_20240330074040_53e10728-cf9a-4a43-9182-1539ddaf78a5  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0002, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0002/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0002  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 07:41:29,304 Stage-1 map = 0%, reduce = 0%  
2024-03-30 07:41:57,756 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 11.21 s  
ec  
2024-03-30 07:42:21,786 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 18.68  
sec  
MapReduce Total cumulative CPU time: 18 seconds 680 msec  
Ended Job = job_1711798292820_0002  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 18.68 sec HDFS Read: 842276  
1 HDFS Write: 10 SUCCESS  
Total MapReduce CPU Time Spent: 18 seconds 680 msec  
OK  
1901 1932
```

proses pencarian nilai maksimum dan minimum tahun

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT MIN(suhu), MAX(suhu) FROM suhu;  
Query ID = oracle_20240330074343_f3076dcd-826c-43d3-b934-079445bcac94  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0003, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0003/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0003  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 07:44:25,546 Stage-1 map = 0%, reduce = 0%  
2024-03-30 07:44:52,255 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 8.3 sec  
2024-03-30 07:45:15,774 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 14.14  
sec  
MapReduce Total cumulative CPU time: 14 seconds 140 msec  
Ended Job = job_1711798292820_0003  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 14.14 sec HDFS Read: 842275  
7 HDFS Write: 6 SUCCESS  
Total MapReduce CPU Time Spent: 14 seconds 140 msec  
OK  
0 999  
Time taken: 102.111 seconds, Fetched: 1 row(s)
```

proses pencarian nilai maksimum suhu


```

hive> SELECT MIN(kualitas), MAX(kualitas) from suhu;
Query ID = oracle_20240330074747_bclclb82-49a3-4f7d-8ca9-b4730b49fc77
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 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_1711798292820_0004, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1711798292820_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-03-30 07:48:34,746 Stage-1 map = 0%, reduce = 0%
2024-03-30 07:49:14,213 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 13.25 s
ec
2024-03-30 07:49:43,630 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 21.58
sec
MapReduce Total cumulative CPU time: 21 seconds 580 msec
Ended Job = job_1711798292820_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 21.58 sec HDFS Read: 842275
9 HDFS Write: 4 SUCCESS
Total MapReduce CPU Time Spent: 21 seconds 580 msec
OK
1 9

```

proses pencarian nilai maksimum dan minimum kualitas

Dilakukan proses pencarian nilai maksimum dan minimum dari tahun, suhu, dan kualitas menggunakan data dari tabel “suhu”. Dengan fungsi MAX(nama_kolom) untuk mencari nilai maksimum dan MIN(nama_kolom) untuk mencari nilai minimum. Diperoleh nilai tahun maksimum 1932 dan nilai minimum sebesar 1901. Diperoleh nilai maksimum dari suhu direntang tahun 1901 sampai 1932 sebesar 999 dan nilai minimum sebesar 0. Dan, diperoleh nilai maksimum kualitas direntang tahun 1901 sampai 1932 sebesar 9 dan nilai minimum sebesar 1.

1.2.2 Nilai Rata-Rata Dari Suhu Dan Kualitas

```

oracle@bigdatalite:~
File Edit View Search Terminal Help
hive> SELECT AVG(suhu) from suhu;
Query ID = oracle_20240330075353_612a4220-b61c-441e-8356-45c7e4316a07
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks determined at compile time: 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_1711798292820_0006, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1711798292820_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2024-03-30 07:54:53,115 Stage-1 map = 0%, reduce = 0%
2024-03-30 07:55:19,371 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 9.87 se
c
2024-03-30 07:55:43,369 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 17.16
sec
MapReduce Total cumulative CPU time: 17 seconds 160 msec
Ended Job = job_1711798292820_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 17.16 sec HDFS Read: 842265
2 HDFS Write: 19 SUCCESS
Total MapReduce CPU Time Spent: 17 seconds 160 msec
OK
141.38859666034392

```

proses pencarian nilai rata-rata suhu

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT AVG(kualitas) from suhu;  
Query ID = oracle_20240330075656_8f2339df-c381-4fe0-bec8-2f1d14759d7e  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0007, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0007/  
Kill Command = /usr/lib/hadoop/bin/hadoop_job -kill job_1711798292820_0007  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 07:57:52,523 Stage-1 map = 0%, reduce = 0%  
2024-03-30 07:58:18,499 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 11.95 s  
ec  
2024-03-30 07:58:41,183 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 17.99  
sec  
MapReduce Total cumulative CPU time: 17 seconds 990 msec  
Ended Job = job_1711798292820_0007  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 17.99 sec HDFS Read: 842266  
0 HDFS Write: 19 SUCCESS  
Total MapReduce CPU Time Spent: 17 seconds 990 msec  
OK  
1.3526490251562548
```

proses pencarian nilai rata-rata kualitas

Dilakukan proses pencarian nilai rata-rata dari suhu dan kualitas menggunakan data dari tabel “suhu” direntang tahun 1901 sampai dengan 1932. Dengan fungsi AVG (nama_kolom) untuk mencari rata-rata untuk mengetahui keseluruhan memahami suhu dan kualitas dalam rentang waktu tersebut. Diperoleh nilai rata-rata suhu sebesar 141,3886 dan nilai rata-rata kualitas sebesar 1,3526,

1.2.3 Nilai Varian Dari Suhu Dan Kualitas

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT VARIANCE(suhu) FROM suhu;  
Query ID = oracle_20240330085555_b1897be8-fffa-43b7-8c18-205dd6ac33e9  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0017, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0017/  
Kill Command = /usr/lib/hadoop/bin/hadoop_job -kill job_1711798292820_0017  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 08:57:06,443 Stage-1 map = 0%, reduce = 0%  
2024-03-30 08:57:31,732 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 8.38 se  
c  
2024-03-30 08:57:55,645 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 15.09  
sec  
MapReduce Total cumulative CPU time: 15 seconds 90 msec  
Ended Job = job_1711798292820_0017  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 15.09 sec HDFS Read: 842268  
9 HDFS Write: 18 SUCCESS  
Total MapReduce CPU Time Spent: 15 seconds 90 msec  
OK  
38495.78147063563
```

proses pencarian nilai varian suhu

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT VARIANCE(kualitas) FROM suhu;  
Query ID = oracle_20240330090101_6ada4f43-d50c-48f2-9428-d3d27d2a0196  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0018, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0018/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0018  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 09:02:44,177 Stage-1 map = 0%, reduce = 0%  
2024-03-30 09:03:14,944 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 14.36 s  
ec  
2024-03-30 09:03:39,150 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 22.48  
sec  
MapReduce Total cumulative CPU time: 22 seconds 480 msec  
Ended Job = job_1711798292820_0018  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 22.48 sec HDFS Read: 842269  
7 HDFS Write: 19 SUCCESS  
Total MapReduce CPU Time Spent: 22 seconds 480 msec  
OK  
2.6931305024889483
```

proses pencarian nilai varian kualitas

Dilakukan proses pencarian nilai varian dari suhu dan kualitas menggunakan data dari tabel “suhu” direntang tahun 1901 sampai dengan 1932. Dengan fungsi VARIANCE(nama_kolom) untuk mencari varians untuk mengetahui nilai sebaran data suhu dan kualitas dalam rentang waktu tersebut. Diperoleh nilai varians suhu sebesar 38495,7815 dan nilai varians kualitas sebesar 2,6931.

1.2.4 Nilai Standar Deviasi Dari Suhu Dan Kualitas

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT STDDEV_POP(suhu) from suhu;  
Query ID = oracle_20240330080000_ebc88573-762c-4d27-b09c-99dcb2332712  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0008, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0008/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0008  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 08:01:01,135 Stage-1 map = 0%, reduce = 0%  
2024-03-30 08:01:25,683 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 8.81 s  
ec  
2024-03-30 08:01:47,091 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 16.15  
sec  
MapReduce Total cumulative CPU time: 16 seconds 150 msec  
Ended Job = job_1711798292820_0008  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 16.15 sec HDFS Read: 842267  
9 HDFS Write: 19 SUCCESS  
Total MapReduce CPU Time Spent: 16 seconds 150 msec  
OK  
196.20341860078696
```

proses pencarian nilai standar deviasi suhu

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT STDDEV_POP(kualitas) from suhu;  
Query ID = oracle_20240330080202_a601fd70-360a-4cf5-a0b0-12d9fd33e0b7  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0009, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0009/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0009  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 08:03:37,594 Stage-1 map = 0%, reduce = 0%  
2024-03-30 08:04:05,408 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 12.29 s  
ec  
2024-03-30 08:04:29,560 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 19.1  
sec  
MapReduce Total cumulative CPU time: 19 seconds 100 msec  
Ended Job = job_1711798292820_0009  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 19.1 sec HDFS Read: 8422687  
HDFS Write: 19 SUCCESS  
Total MapReduce CPU Time Spent: 19 seconds 100 msec  
OK  
1.6410760197166212
```

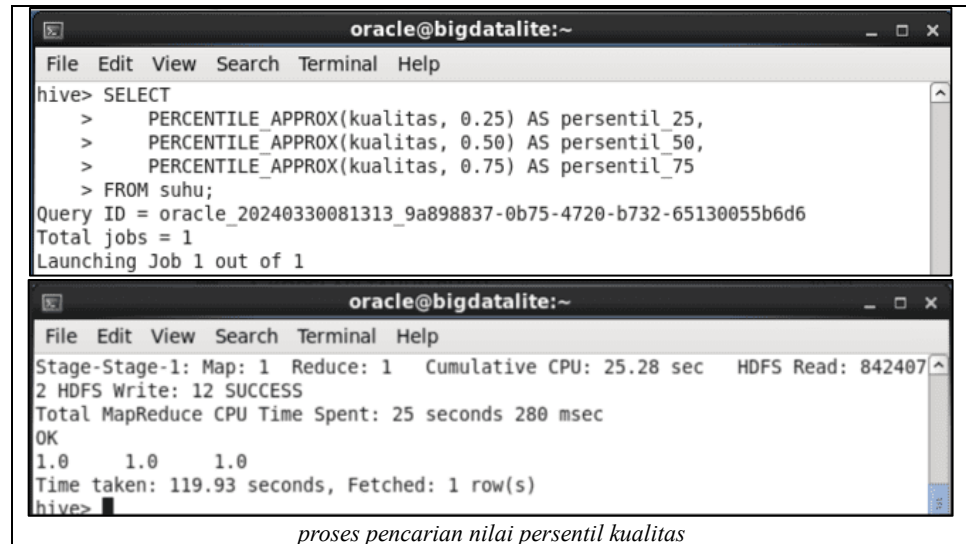
proses pencarian nilai standar deviasi kualitas

Dilakukan proses pencarian nilai standar deviasi dari suhu dan kualitas menggunakan data dari tabel “suhu” direntang tahun 1901 sampai dengan 1932. Dengan fungsi STDDEV_POP(nama_kolom) menggunakan populasi karena jumlah data lebih dari 30 untuk mencari nilai standar deviasi untuk mengetahui seberapa jauh nilai tersebar dari nilai rata-rata dalam rentang waktu tersebut. Diperoleh nilai standar deviasi suhu sebesar 196,2034 dan nilai standar deviasi kualitas sebesar 1,6411.

1.2.5 Nilai Persentil Dari Suhu Dan Kualitas

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT  
  > PERCENTILE_APPROX(suhu, 0.25) AS persentil_25,  
  > PERCENTILE_APPROX(suhu, 0.50) AS persentil_50,  
  > PERCENTILE_APPROX(suhu, 0.75) AS persentil_75  
  > FROM suhu;  
Query ID = oracle_20240330090505_6c62edc3-3b65-4cdb-8ad2-12d15e6d302b  
Total jobs = 1  
Launching Job 1 out of 1  
  
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 15.36 sec HDFS Read: 842403  
6 HDFS Write: 56 SUCCESS  
Total MapReduce CPU Time Spent: 15 seconds 360 msec  
OK  
48.305011202625515      98.37210684567513      155.17353765770358  
Time taken: 130.496 seconds, Fetched: 1 row(s)  
hive>
```

proses pencarian nilai persentil suhu



The image shows two screenshots of a terminal window titled 'oracle@bigdatalite:~'. The top screenshot shows a Hive query being executed: `hive> SELECT > PERCENTILE_APPROX(kualitas, 0.25) AS persentil_25, > PERCENTILE_APPROX(kualitas, 0.50) AS persentil_50, > PERCENTILE_APPROX(kualitas, 0.75) AS persentil_75 > FROM suhu;` The output shows 'Query ID = oracle_20240330081313_9a898837-0b75-4720-b732-65130055b6d6', 'Total jobs = 1', and 'Launching Job 1 out of 1'. The bottom screenshot shows the execution progress: 'Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 25.28 sec HDFS Read: 842407', '2 HDFS Write: 12 SUCCESS', 'Total MapReduce CPU Time Spent: 25 seconds 280 msec', 'OK', and 'Time taken: 119.93 seconds, Fetched: 1 row(s)'. The terminal prompt 'hive>' is visible at the bottom of both screenshots.

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT  
> PERCENTILE_APPROX(kualitas, 0.25) AS persentil_25,  
> PERCENTILE_APPROX(kualitas, 0.50) AS persentil_50,  
> PERCENTILE_APPROX(kualitas, 0.75) AS persentil_75  
> FROM suhu;  
Query ID = oracle_20240330081313_9a898837-0b75-4720-b732-65130055b6d6  
Total jobs = 1  
Launching Job 1 out of 1  
  
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 25.28 sec HDFS Read: 842407  
2 HDFS Write: 12 SUCCESS  
Total MapReduce CPU Time Spent: 25 seconds 280 msec  
OK  
1.0 1.0 1.0  
Time taken: 119.93 seconds, Fetched: 1 row(s)  
hive>
```

proses pencarian nilai persentil kualitas

Dilakukan proses pencarian nilai persentil dari suhu dan kualitas menggunakan data dari tabel “suhu” direntang tahun 1901 sampai dengan 1932. Dengan fungsi `PERCENTILE_APPROX(nama_kolom, nilai_yang_ditentukan)`. Dilakukan pencarian nilai persentil ke-25, 50, dan 75. Diperoleh nilai persentil 25 dari suhu sebesar 48,3035 merepresentasikan suhu memiliki nilai lebih rendah dari nilai persentil 25, persentil 50 sebesar 98,3721 merepresentasikan 50% data memiliki nilai lebih rendah dan 50% lainnya memiliki nilai lebih tinggi, dan persentil 75 sebesar 155,1735 merepresentasikan nilai suhu berada dibawah nilai persentil 75.

Dan dilakukan pencarian nilai persentil ke-25, 50, dan 75. Diperoleh nilai persentil 25 dari kualitas sebesar 1 persentil 50 sebesar 1, dan persentil 75 sebesar 1.

2. Analisis Kenaikan Suhu Di Antara Dua Tahun

Dilakukan proses analisis kenaikan suhu di antara dua tahun untuk mengetahui apakah pola perubahan iklim dan mengetahui dampak signifikan dari perubahan iklim. Pada proses analisis ini dilakukan analisis kenaikan suhu di antara dua tahun dalam rentang waktu 1901 sampai dengan 1932. Dimana untuk melakukan analisis ini dipergunakan perhitungan nilai rata-rata suhu disetiap tahunnya.

Dengan menggunakan fungsi LAG diperhitungkan nilai rata-rata suhu pertahun mulai tahun 1901 sampai 1932 kemudian dilakukan perhitungan selisih untuk mengetahui apakah di antara satu tahun saat ini dengan tahun sebelumnya terjadi kenaikan suhu atau penurunan suhu. *Output* tersebut akan disimpan terlebih dahulu didalam HIVE dengan nama “analisis_kenaikan_suhu” dan dilakukan pemindahan *file output* ke local menggunakan `-getmerge`. *Link output keseluruhan :*

https://drive.google.com/file/d/1fEFAW-KqFsWzqm9IB5-Ao2wh9zR-bJIj/view?usp=drive_link


```

oracle@bigdatalite:~
File Edit View Search Terminal Help
hive> INSERT OVERWRITE DIRECTORY '/user/hive/analisis_kenaikan_suhu'
> ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'
> SELECT
>   tahun,
>   rataratasuhu,
>   LAG(rataratasuhu) OVER (ORDER BY tahun) AS rataratasuhu_sebelumnya,
>   rataratasuhu - LAG(rataratasuhu) OVER (ORDER BY tahun) AS kenaikan_suh
u
> FROM (
>   SELECT
>     tahun,
>     AVG(suhu) AS rataratasuhu
>   FROM
>     suhu
>   GROUP BY
>     tahun
> ) AS rata_tahun;
Query ID = oracle_20240330083737_cb3ed790-29d7-43e2-80e3-cc098a5a05f5
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1

```

proses pencarian kenaikan suhu di antara dua tahun

```

oracle@bigdatalite:~
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ hadoop fs -getmerge /user/hive/analisis_kenaikan_suhu an
alisis_kenaikan_suhu.txt

```

proses pemindahan file output ke lokal

analisis_kenaikan_suhu.txt (~) - gedit

1901	93.8994668697639	\N	\N
1902	73.78385376999239	93.8994668697639	-20.115613099771508
1903	78.52105584375954	73.78385376999239	4.7372020737671505
1904	74.4364934670313	78.52105584375954	-4.084562376728243
1905	75.15333028501753	74.4364934670313	0.7168368179862341
1906	83.6172816952868	75.15333028501753	8.463951410269274
1907	89.30807248764415	83.6172816952868	5.6907907923573475
1908	84.8540622627183	89.30807248764415	-4.454010224925852
1909	91.83727380795139	84.8540622627183	6.98321154523309
1910	78.11803921568628	91.83727380795139	-13.71923459226511
1911	86.5815352425788	78.11803921568628	8.463496026892514
1912	92.43987506507028	86.5815352425788	5.858339822491487
1913	85.17100753941055	92.43987506507028	-7.268867525659729
1914	80.78346861781182	85.17100753941055	-4.387538921598733
1915	91.46176705909247	80.78346861781182	10.678298441280646
1916	62.619004111466424	91.46176705909247	-28.842762947626042
1917	94.20697541452259	62.619004111466424	31.587971303056165
1918	86.09712837837837	94.20697541452259	-8.109847036144217

proses penampilan file output

Output yang ditampilkan terdapat empat kolom dengan kolom 1 merupakan tahun, kolom 2 merupakan nilai rata-rata suhu pada tahun tersebut, kolom 3 merupakan nilai rata-rata suhu pada tahun sebelumnya, dan kolom 4 merupakan hasil analisis kenaikan suhu dengan nilai positif dan penurunan suhu dengan nilai negatif. Dimana untuk melakukan analisis peningkatan atau penurunan suhu dapat dipergunakan rumus.

Analisis Kenaikan Atau Penurunan Suhu :

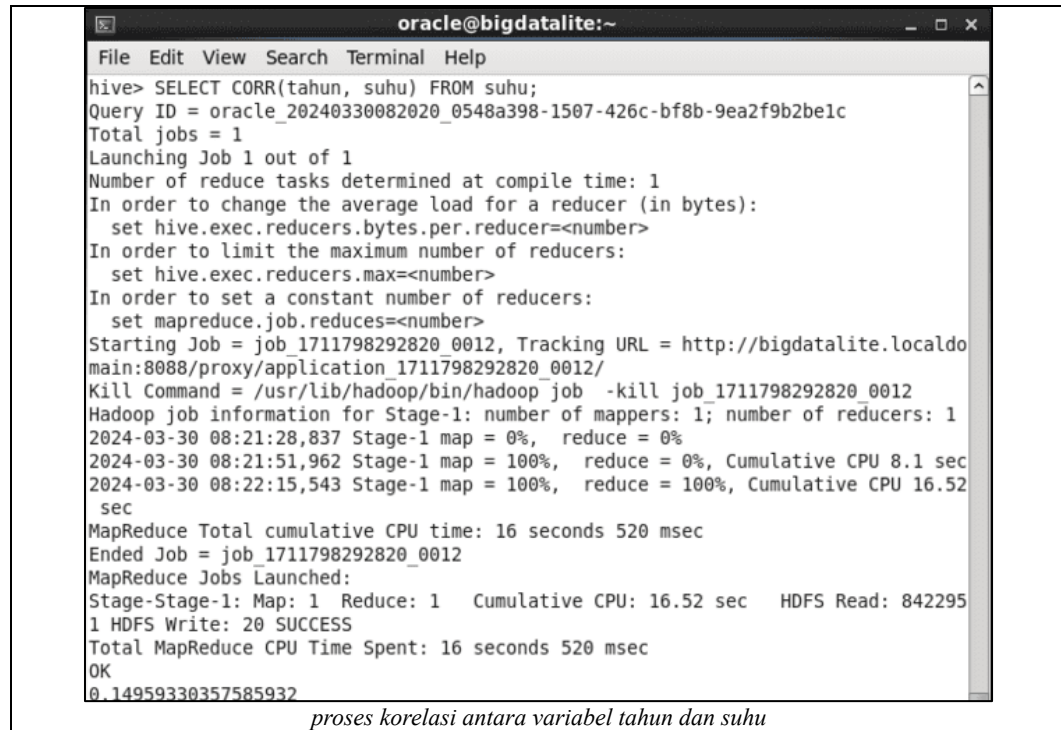
Nilai ratarata suhu tahun ini – Nilai ratarata suhu tahun sebelumnya

3. Korelasi Antara 2 Variabel (Korelasi Positif/Korelasi Negatif/Tidak Berkorelasi)

Dilakukan proses analisis korelasi antar 2 variabel untuk memahami hubungan atau keterkaitan antara kedua variabel tersebut. Dengan korelasi positif memiliki nilai direntang 0 hingga 1 yang menunjukkan ketika satu variabel naik, maka variabel lainnya juga naik. Korelasi negatif memiliki nilai direntang -1 hingga 0 yang menunjukkan ketika satu variabel naik, maka variabel lainnya turun.

Dan, tidak berkorelasi yang memiliki nilai 0 yang menunjukkan bahwa tidak ada hubungan yang sistematis diantara dua variabel. Pada proses ini dilakukan korelasi antara dua variabel, dengan variabel tahun dan suhu, variabel tahun dan kualitas, dan variabel suhu dan kualitas.

3.1 Korelasi Variabel Tahun Dan Suhu



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT CORR(tahun, suhu) FROM suhu;  
Query ID = oracle_20240330082020_0548a398-1507-426c-bf8b-9ea2f9b2be1c  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0012, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0012/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0012  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 08:21:28,837 Stage-1 map = 0%, reduce = 0%  
2024-03-30 08:21:51,962 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 8.1 sec  
2024-03-30 08:22:15,543 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 16.52  
sec  
MapReduce Total cumulative CPU time: 16 seconds 520 msec  
Ended Job = job_1711798292820_0012  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 16.52 sec HDFS Read: 842295  
1 HDFS Write: 20 SUCCESS  
Total MapReduce CPU Time Spent: 16 seconds 520 msec  
OK  
0.14959330357585932
```

proses korelasi antara variabel tahun dan suhu

Dilakukan proses korelasi untuk mengetahui hubungan antara variabel tahun dan suhu menggunakan data dari tabel “suhu”. Dengan fungsi CORR(namakolom1, namakolom2). Dilakukan pencarian nilai korelasi dengan CORR(tahun, suhu) dan diperoleh nilai *output* korelasi sebesar 0,1496 yang merepresentasikan adanya korelasi positif yang cenderung lemah, dengan seiring bertambahnya tahun maka suhu juga cenderung bertambah atau meningkat.

3.2 Korelasi Variabel Tahun Dan Kualitas

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT CORR(tahun, kualitas) FROM suhu;  
Query ID = oracle_20240330082323_3e32bd3f-3cd2-49ba-aaef-daa5490d5b9d  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 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_1711798292820_0013, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0013/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0013  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 08:24:01,319 Stage-1 map = 0%, reduce = 0%  
2024-03-30 08:24:31,819 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 15.11 s  
ec  
2024-03-30 08:24:53,354 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 22.23  
sec  
MapReduce Total cumulative CPU time: 22 seconds 230 msec  
Ended Job = job_1711798292820_0013  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 22.23 sec HDFS Read: 842296  
0 HDFS Write: 20 SUCCESS  
Total MapReduce CPU Time Spent: 22 seconds 230 msec  
OK  
0.10951971051180272
```

proses korelasi antara variabel tahun dan kualitas

Dilakukan proses korelasi untuk mengetahui hubungan antara variabel tahun dan kualitas menggunakan data dari tabel “suhu”. Dengan fungsi CORR(namakolom1, namakolom2). Dilakukan pencarian nilai korelasi dengan CORR(tahun, kualitas) dan diperoleh nilai *output* korelasi sebesar 0,1095 yang merepresentasikan adanya korelasi positif yang cenderung lemah, dengan seiring bertambahnya tahun maka kualitas juga cenderung bertambah atau meningkat.

3.3 Korelasi Variabel Suhu Dan Kualitas

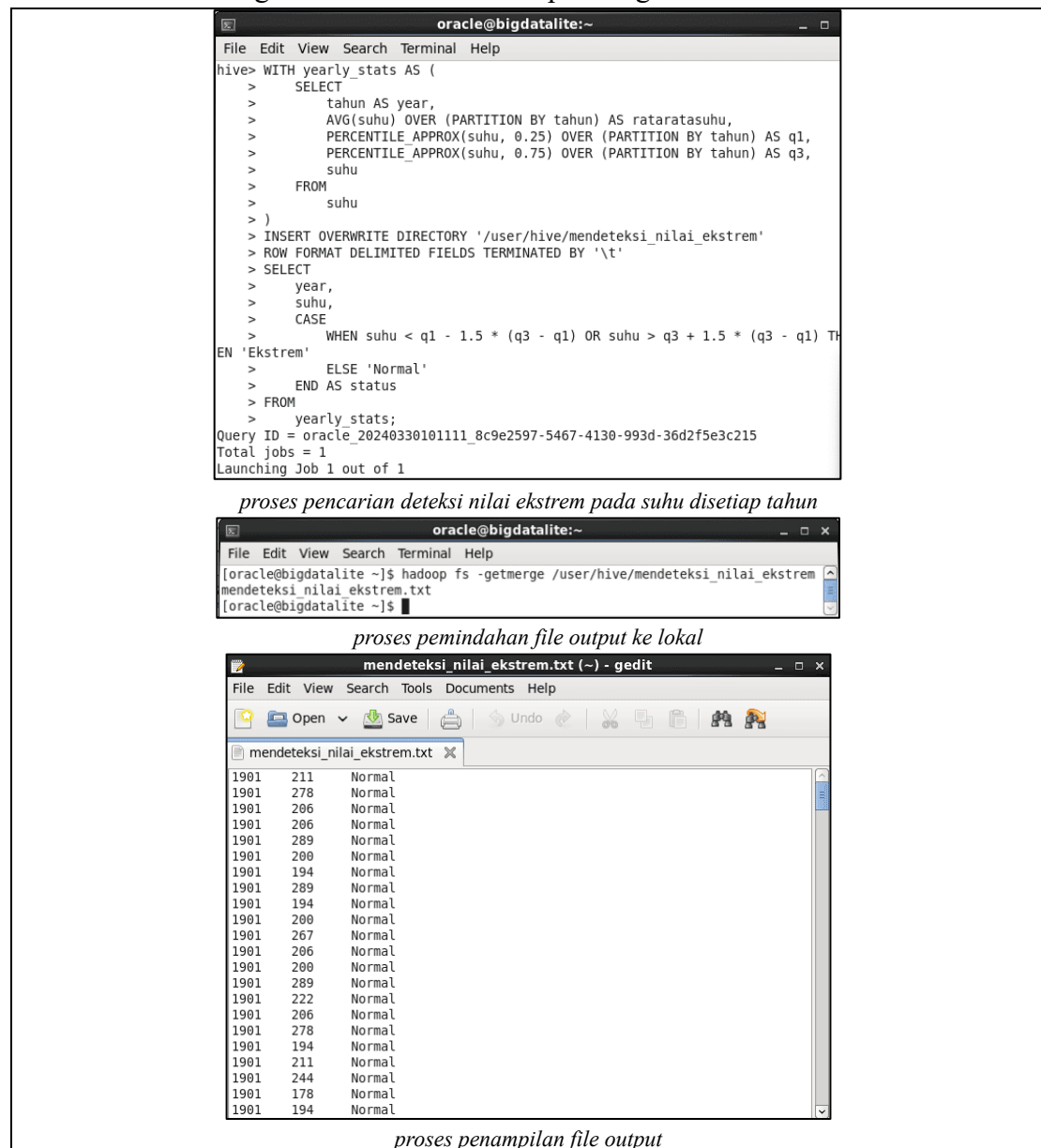
```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
hive> SELECT CORR(suhu, kualitas) FROM suhu;  
Query ID = oracle_20240330082525_97955144-2b5d-4655-a821-67fa9848074d  
Total jobs = 1  
Launching Job 1 out of 1  
Number of reduce tasks determined at compile time: 1  
In order to change 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_1711798292820_0014, Tracking URL = http://bigdatalite.localdo  
main:8088/proxy/application_1711798292820_0014/  
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1711798292820_0014  
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1  
2024-03-30 08:26:12,702 Stage-1 map = 0%, reduce = 0%  
2024-03-30 08:26:46,972 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 15.87 s  
ec  
2024-03-30 08:27:08,947 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 22.65  
sec  
MapReduce Total cumulative CPU time: 22 seconds 650 msec  
Ended Job = job_1711798292820_0014  
MapReduce Jobs Launched:  
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 22.65 sec HDFS Read: 842295  
8 HDFS Write: 19 SUCCESS  
Total MapReduce CPU Time Spent: 22 seconds 650 msec  
OK  
0.9378911247187945
```

proses korelasi antara variabel suhu dan kualitas

Dilakukan proses korelasi untuk mengetahui hubungan antara variabel tahun dan kualitas menggunakan data dari tabel “suhu”. Dengan fungsi CORR(namakolom1, namakolom2). Dilakukan pencarian nilai korelasi dengan CORR(suhu, kualitas) dan diperoleh nilai *output* korelasi sebesar 0,9389 yang merepresentasikan adanya korelasi positif kuat, dengan seiring bertambahnya suhu maka kualitas juga bertambah atau meningkat.

4. Deteksi Nilai-Nilai Ekstrem Pada Suhu Di Tiap-Tiap Tahun

Dilakukan proses analisis deteksi nilai-nilai ekstrem pada suhu disetiap tahunnya untuk mengetahui tahun dimana suhu mencapai nilai yang ekstrem baik tinggi ataupun rendah dalam rentang waktu tahun 1901 sampai dengan 1932.



The figure illustrates the process of detecting extreme temperature values through three sequential steps:

- proses pencarian deteksi nilai ekstrem pada suhu disetiap tahun**: A terminal window showing a Hive query that calculates the average temperature and the 25th and 75th percentiles for each year, then identifies years where the temperature is significantly below or above these percentiles as 'Ekstrem'.
- proses pemindahan file output ke lokal**: A terminal window showing the execution of a Hadoop command to retrieve the output file from the HDFS storage.
- proses penampilan file output**: A text editor window displaying the output file, which lists years and their corresponding temperature status (Normal or Ekstrem).

Tahun	Suhu	Status
1901	211	Normal
1901	278	Normal
1901	206	Normal
1901	206	Normal
1901	289	Normal
1901	200	Normal
1901	194	Normal
1901	289	Normal
1901	194	Normal
1901	200	Normal
1901	267	Normal
1901	206	Normal
1901	200	Normal
1901	289	Normal
1901	222	Normal
1901	206	Normal
1901	278	Normal
1901	194	Normal
1901	211	Normal
1901	244	Normal
1901	178	Normal
1901	194	Normal

Dengan melakukan analisis statistik tahunan terhadap data suhu dengan fokus variabel kolom suhu dan tahun dilakukan analisis perhitungan rata-rata menggunakan

fungsi AVG dan nilai kuartil pertama dan kuartil ketiga menggunakan fungsi PERCENTILE_APPROX. Kemudian, dilakukan pembuatan kondisi dengan suhu dianggap “Ekstrem” jika melewati batasan tertentu yang ditentukan berdasarkan kuartil pertama dan kuartil ketiga.

Dengan jika nilainya kurang dari kuartil pertama dikurangi 1,5 kali rentang interkuartil atau lebih besar dari kuartil tiga ditambah 1,5 kali rentang interkuartil. Jika tidak memenuhi kondisi tersebut, maka nilai suhu dianggap normal. *Output* tersebut akan disimpan terlebih dahulu didalam HIVE dengan nama “mendeteksi_nilai_ekstrem” dan dilakukan pemindahan *file output* ke local menggunakan *-getmerge*. *Link ouput keseluruhan* :

https://drive.google.com/file/d/1Mk7_n7COYud3DEJxgAp0Vc_wpmMaui6v/view?usp=drive_link

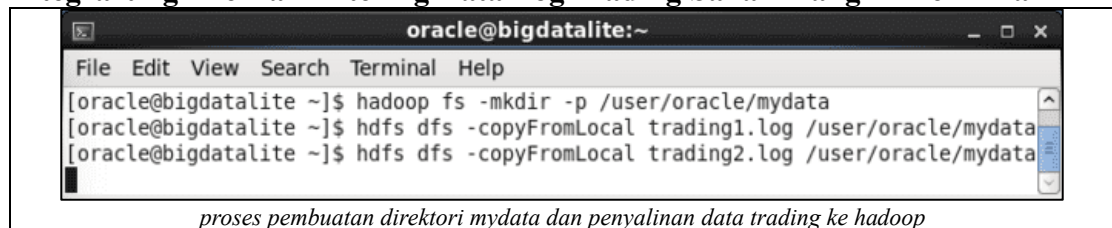
B. Data Log Trading Saham

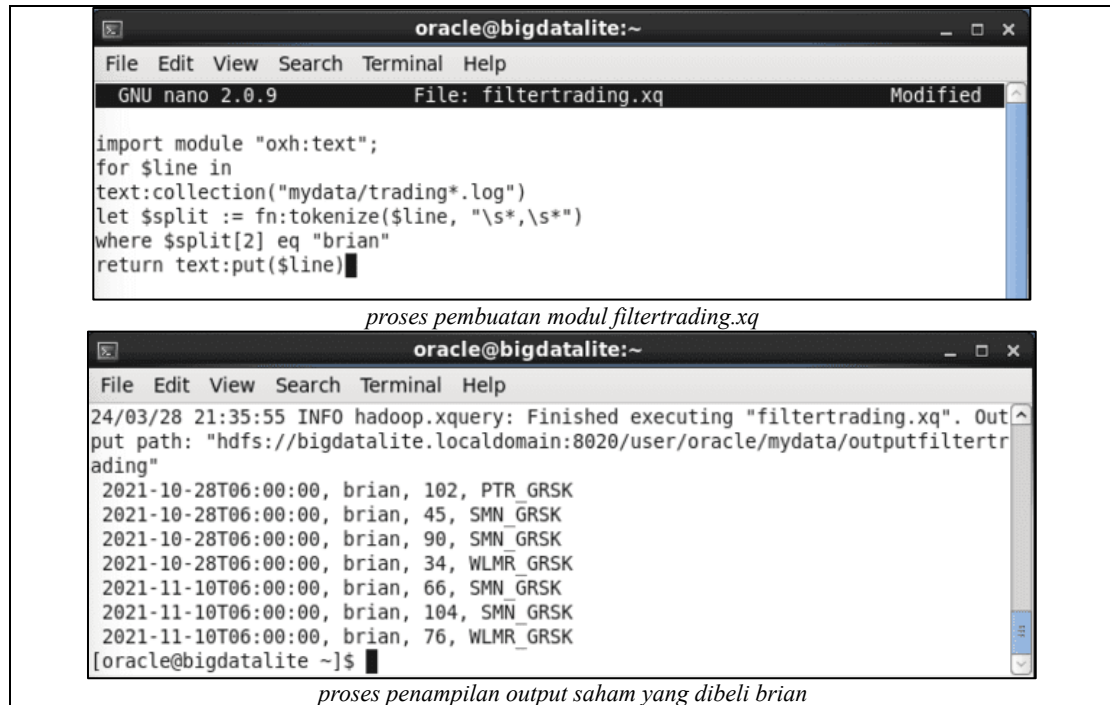
Data log trading saham merupakan data yang merepresentasikan catatan atau rekaman dari aktivitas perdagangan saham yang terjadi di pasar keuangan. Dimana pada data log trading saham ini terbagi menjadi dua, yakni data trader.txt yang berisi nama panggilan orang yang membeli saham, umur, dan nama lengkapnya. Kemudian terdapat data trading1.log dan trading2.log yang berisi tanggal pembelian, nama yang membeli saham, jumlah, dan nama perusahaan yang dibeli sahamnya.

Dengan menggunakan data log trading saham akan dilakukan proses penerapan XQuery Transformation. XQuery Transformation merupakan konsep pemrosesan data yang mengacu pada ekspresi atau perubahan yang diterapkan pada kueri data. Dengan melibatkan berbagai jenis operasi, seperti penggabungan (*join*), penyaringan (*filter*), pengelompokan (*grouping*), dan lain sebagainya. Bertujuan untuk mengubah struktur atau nilai data kedalam bentuk kueri untuk mendapatkan informasi mengenai analisis tertentu.

Pada data log trading saham, dilakukan beberapa penerapan XQuery Transformation, meliputi proses *integrating files*, *basic filtering*, *group by and aggregation*, *inner joins*, *left outer joins*, *semi joins*, *multiple outputs*, *accessing auxiliary input data*, *custom java function from XQuery*, dan *user-defined XQuery library modules and XML schemas* yang akan dijelaskan lebih lanjut sebagai berikut.

1. Integrating File Dan Filtering Data Log Trading Saham Yang Di Beli Brian

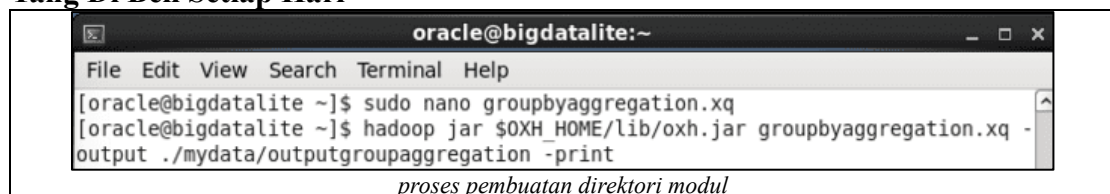


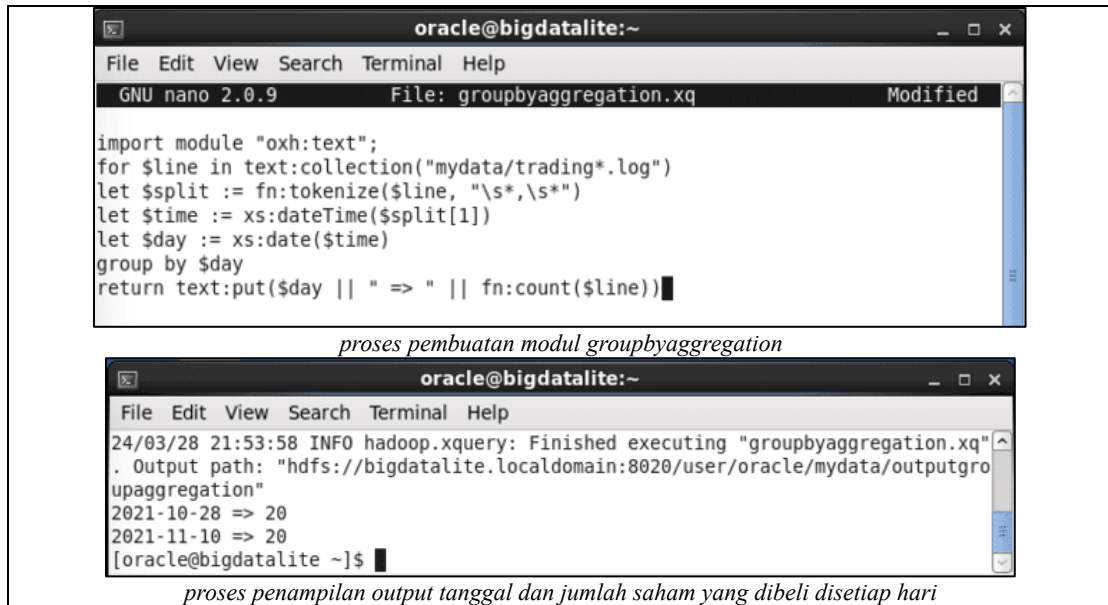


Dilakukan penerapan penggabungan *file* dan *filtering* dengan menggunakan *argument* yang ditulis dalam modul “*\$sudo nano filtertrading.xq*” dengan modul bernama, a “*filteringtrading.xq*”. Dilakukan proses pembuatan direktori bernama “*mydata*” dan menyalin *file* data *trading1.log* dan *trading2.log* kedalam *hadoop*. Kemudian, dilakukan proses penggabungan data *trading1.log* dan *trading2.log* menggunakan *argument* yang ditulis di dalam modul, yakni “*text:collection(“mydata/trading*.log”)*”. Setelah melakukan penggabungan data *trading1.log* dan *trading2.log*, langkah selanjutnya adalah melakukan proses *filtering* untuk penampilan transaksi pembelian saham oleh seorang *trader*.

Kemudian, dilakukan proses *filtering* atau pemilihan pembelian saham yang dilakukan oleh “*brian*” dengan menggunakan *argument* yang ditulis didalam modul, yakni “*where \$split[2] eq “brian”*”. Dan, ditampilkan *output* sebanyak 7 saham yang dibeli *brian* dengan informasi tanggal pembelian, harga, dan nama perusahaan saham yang dibeli.

2. Group By And Aggregation Untuk Menampilkan Tanggal Dan Jumlah Saham Yang Di Beli Setiap Hari





The first screenshot shows a terminal window with the nano editor open to create a file named `groupbyaggregation.xq`. The code inside is as follows:

```
import module "oxh:text";
for $line in text:collection("mydata/trading*.log")
let $split := fn:tokenize($line, "\s*,\s*")
let $time := xs:dateTime($split[1])
let $day := xs:date($time)
group by $day
return text:put($day || " => " || fn:count($line))
```

The second screenshot shows the terminal output after running the module. It displays the execution time, the output path, and the results of the aggregation:

```
24/03/28 21:53:58 INFO hadoop.xquery: Finished executing "groupbyaggregation.xq"
. Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputgro
upaggregation"
2021-10-28 => 20
2021-11-10 => 20
[oracle@bigdatalite ~]$
```

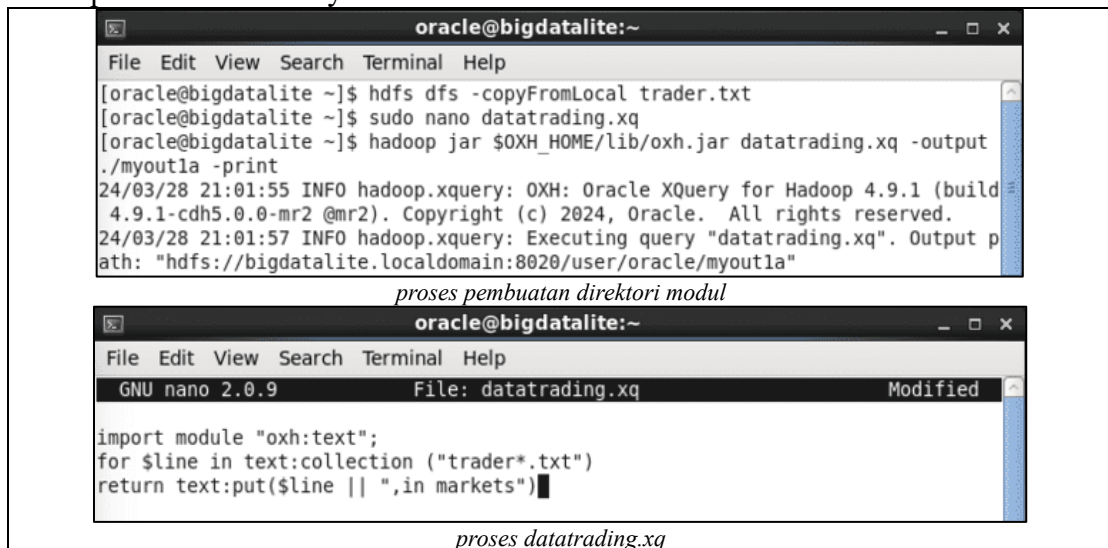
proses pembuatan modul groupbyaggregation

proses penampilan output tanggal dan jumlah saham yang dibeli disetiap hari

Dilakukan penerapan *group by aggregation* untuk menampilkan tanggal dan jumlah saham yang dibeli berdasarkan data pada `trading1.log` dan `trading2.log`. Dengan melakukan pembuatan modul yang bernama “`groupbyaggregation.xq`” dengan *argument* menggunakan semua data trading dan melakukan pengambilan data kolom pertama yang diubah formatnya menjadi tanggal. Kemudian, dilakukan pengelompokkan berdasarkan tanggal, dan ditampilkan *output* pada tanggal 28 bulan 10 tahun 2021 dilakukan pembelian saham sebanyak 20 serta pada tanggal 10 bulan 11 tahun 2021 dilakukan pembelian saham sebanyak 20.

3. Inner Joins Untuk Penampilan Jumlah Pembelian Saham Di Tiap Perusahaan Yang Berusia Lebih Dari 25 Tahun

Inner join merupakan jenis operasi penggabungan (*join*) antara dataset yang tersimpan dalam hadoop. Perlu diketahui bahwa sebelumnya telah melakukan penggabungan data `trading1.log` dan `trading2.log`. Kemudian, akan dilakukan proses penyalinan data “`trader.txt`” ke dalam hadoop dan membuat modul “`datatrading.xq`” untuk menampilkan *output* dari `trader.txt` dengan penambahan kalimat “*in markets*” disetiap akhir baris datanya.



The first screenshot shows a terminal window with the following commands being executed:

```
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal trader.txt
[oracle@bigdatalite ~]$ sudo nano datatrading.xq
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar datatrading.xq -output
./myoutla -print
24/03/28 21:01:55 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build
4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
24/03/28 21:01:57 INFO hadoop.xquery: Executing query "datatrading.xq". Output p
ath: "hdfs://bigdatalite.localdomain:8020/user/oracle/myoutla"
```

The second screenshot shows the nano editor open to create a file named `datatrading.xq`. The code inside is as follows:

```
import module "oxh:text";
for $line in text:collection ("trader*.txt")
return text:put($line || ",in markets")
```

proses pembuatan direktori modul

proses datatrading.xq


```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/28 21:03:27 INFO hadoop.xquery: Finished executing "datatrading.xq". Output  
path: "hdfs://bigdatalite.localdomain:8020/user/oracle/myout1a"  
indra:26:Indra Kenz,in markets  
fakarich:31:Fakar Suhartami,in markets  
doni:24:Doni Salmanan,in markets  
brian:27:Brian Edgar Nababan,in markets  
[oracle@bigdatalite ~]$
```

proses penampilan nama panggilan, usia, dan nama lengkap pembeli saham

Kemudian, dilakukan proses inner joins dengan pembuatan modul XQuery bernama "innerjoin.xq dengan *argument* menggunakan semua data pada data trader.txt dan trading1.log, trading2.log untuk mencari nama saham perusahaan yang dibeli oleh orang dengan usia lebih dari 25 tahun. Diperoleh bahwa terdapat sebanyak 3 orang yang berusia di atas 25 tahun, meliputi indra, brian, dan fakarich. Kemudian, ketiga orang tersebut memiliki transaksi membeli saham di PTR_GRSK sebanyak 11, SMN_GRSK sebanyak 11, dan WLMR_GRSK sebanyak 6.

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano innerjoin.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar innerjoin.xq -output ./mydata/outerinnerjoin -print  
24/03/29 03:43:56 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
```

proses pembuatan direktori modul

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: innerjoin.xq Modified  
import module "oxh:text";  
  
for $userLine in text:collection("mydata/trader.txt")  
let $userSplit := fn:tokenize($userLine, "\s*:\s*")  
let $userId := $userSplit[1]  
let $userAge := xs:integer($userSplit[2][. castable as xs:integer])  
  
for $visitLine in text:collection("mydata/trading*.log")  
let $visitSplit := fn:tokenize($visitLine, "\s*,\s*")  
let $visitUserId := $visitSplit[2]  
where $userId eq $visitUserId and $userAge gt 25  
group by $page := $visitSplit[4]  
return text:put($page || " " || fn:count($userLine))
```

proses innerjoin.xq

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 03:51:21 INFO hadoop.xquery: Finished executing "innerjoin.xq". Output  
path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outerinnerjoin"  
PTR_GRSK 11  
SMN_GRSK 11  
WLMR_GRSK 6  
[oracle@bigdatalite ~]$
```

proses penampilan nama panggilan, usia, dan nama lengkap

4. Left Outer Join Untuk Mencari Nama Orang Yang Memiliki Saham Ataupun Tidak

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: outerjoin.xq  
  
import module "oxh:text";  
  
for $userLine in text:collection("mydata/trader.txt")  
let $userSplit := fn:tokenize($userLine, "\s*:\s*")  
let $userId := $userSplit[1]  
for $visitLine allowing empty in  
text:collection("mydata/trading*.log")  
[$userId eq fn:tokenize(., "\s*:\s*")[2]]  
  
group by $userId  
return text:put($userId || " " || fn:count($visitLine))  
  
proses outerjoin.xq  
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 04:30:04 INFO hadoop.xquery: Finished executing "outerjoin.xq". Output  
path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputouterjoin"  
brian 7  
doni 12  
fakarich 8  
indra 13  
  
proses penampilan nama orang yang memiliki saham ataupun tidak
```

Left outer join merupakan jenis penggabungan (*join*) untuk menggabungkan data dari dua tabel berdasarkan kondisi yang diberikan dengan mempertahankan semua baris dari tabel kiri, dan memasangkannya pada tabel kanan. Jadi, ketika tidak ada nilai yang cocok di tabel kanan untuk baris tertentu di tabel kiri, maka kolom yang berasal dari tabel kanan akan berisi nilai null. Dilakukan pembuatan modul XQuery Transformation “\$sudo nano outerjoin.xq” bernama “outerjoin.xq”.

Dengan *argument* menggunakan semua data trader.txt di kolom pertama berisi untuk dipasangkan dengan kolom kedua di data trading1.log dan trading2.log berisi nama orang yang mempunyai saham. Ditampilkan hasil bahwa brian memiliki transaksi pembelian saham sebesar 7, doni sebesar 12, fakarich sebesar 8, dan indra sebesar 13.

5. Semi Joins Untuk Mencari Nama Pelanggan Yang Memiliki Saham

```
[oracle@bigdatalite ~]$ sudo nano semijoin.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar semijoin.xq -output ./mydata/outputsemijoin -print  
24/03/29 04:45:59 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.  
  
proses pembuatan direktori modul  
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: semijoin.xq  
  
import module "oxh:text";  
  
for $userLine in text:collection("mydata/trader.txt")  
let $userId := fn:tokenize($userLine, "\s*:\s*")[1]  
  
where some $visitLine in text:collection("mydata/trading*.log")  
satisfies $userId eq fn:tokenize($visitLine, "\s*:\s*")[2]  
return text:put($userId)  
  
proses semijoin.xq
```

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 04:51:49 INFO hadoop.xquery: Finished executing "semijoin.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputsemijoin"  
brian  
doni  
fakarich  
indra  
[oracle@bigdatalite ~]$
```

proses penampilan nama orang yang memiliki saham

Semi join merupakan jenis penggabungan (*join*) yang menghasilkan kumpulan data dari tabel kiri yang memiliki nilai sesuai ditabel kanan. Dilakukan pembuatan modul XQuery Transformation bernama “semijoin.xq” dengan *argument* menggunakan kolom pertama ditabel trader.txt yang berisi nama dan kolom kedua dari data tabel trading1.log dan trading2.log yang berisi nama orang yang memiliki saham. Ditampilkan hasil, bahwa brian, doni, fakarich, dan indra memiliki saham.

6. Multiple Outputs Untuk Melacak Orang (Trader) Yang Memiliki Saham Di “WLMR_GRSK”

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano multipleoutputs.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar multipleoutputs.xq -output ./mydata/outputmultiple -print  
24/03/29 05:52:35 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
```

proses pembuatan direktori modul

```
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: multipleoutputs.xq  
import module namespace text = "oxh:text";  
for $visitLine in text:collection("mydata/trading*.log")  
let $visitCode := fn:tokenize($visitLine, "\s*,\s*")[4]  
return if (fn:contains($visitCode, "WLMR_GRSK")) then text:trace($visitLine) else text:put($visitLine)
```

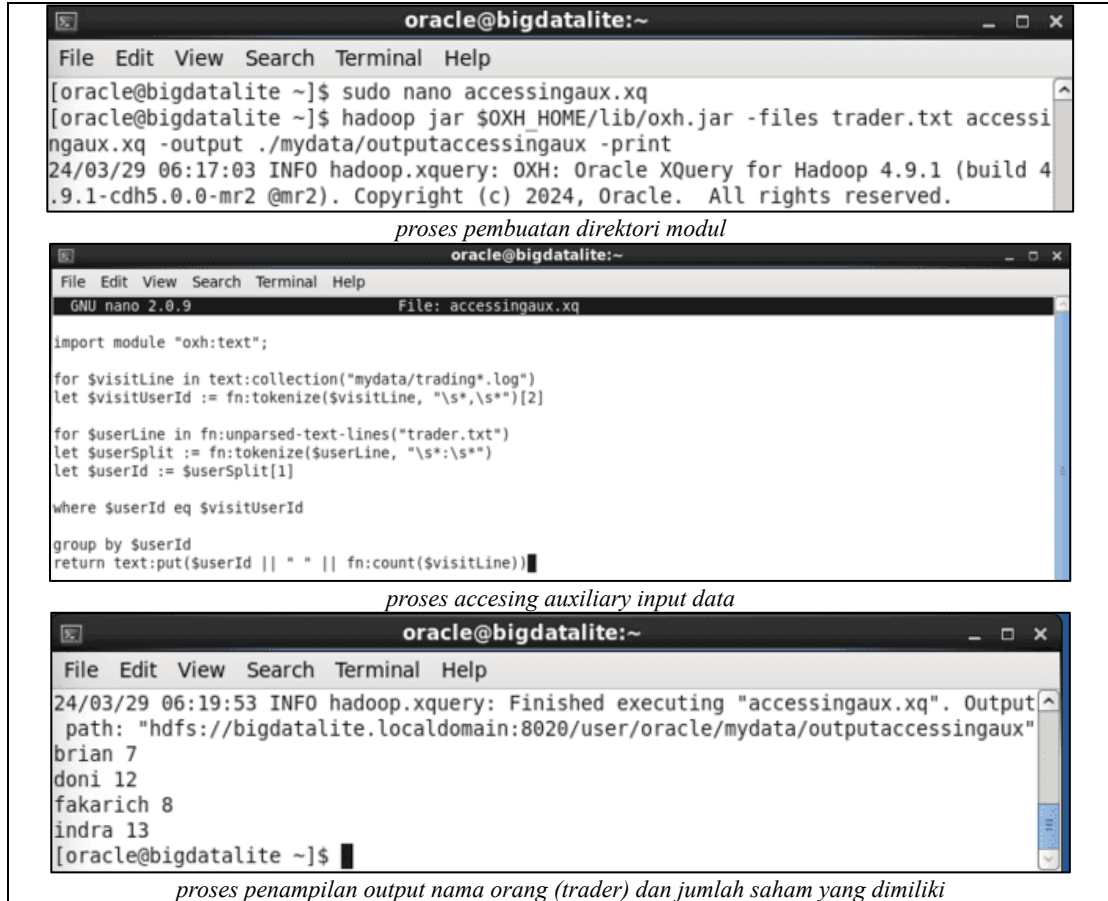
proses multiplesoutputs.xq

```
File Edit View Search Terminal Help  
put path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputmultiple"  
2021-10-28T06:00:00, indra, 100, SPM, GRSK  
2021-10-28T06:00:00, brian, 102, PTR, GRSK  
2021-10-28T06:00:00, fakarich, 90, SPM, GRSK  
2021-10-28T06:00:00, doni, 86, PTR, GRSK  
2021-10-28T06:00:00, indra, 108, PTR, GRSK  
2021-10-28T06:00:00, brian, 45, SPM, GRSK  
2021-10-28T06:00:00, indra, 100, PTR, GRSK  
2021-10-28T06:00:00, doni, 77, SPM, GRSK  
2021-10-28T06:00:00, doni, 67, PTR, GRSK  
2021-10-28T06:00:00, fakarich, 101, PTR, GRSK  
2021-10-28T06:00:00, doni, 56, PTR, GRSK  
2021-10-28T06:00:00, doni, 73, SPM, GRSK  
2021-10-28T06:00:00, indra, 82, PTR, GRSK  
2021-10-28T06:00:00, brian, 99, SPM, GRSK  
2021-10-28T06:00:00, indra, 109, SPM, GRSK  
2021-10-28T06:00:00, fakarich, 110, SPM, GRSK  
2021-11-10T06:00:00, brian, 66, SPM, GRSK  
2021-11-10T06:00:00, fakarich, 90, PTR, GRSK  
2021-11-10T06:00:00, doni, 93, SPM, GRSK  
2021-11-10T06:00:00, indra, 82, PTR, GRSK  
2021-11-10T06:00:00, indra, 103, PTR, GRSK  
2021-11-10T06:00:00, fakarich, 64, SPM, GRSK  
2021-11-10T06:00:00, doni, 91, PTR, GRSK  
2021-11-10T06:00:00, indra, 71, SPM, GRSK  
2021-11-10T06:00:00, indra, 63, PTR, GRSK  
2021-11-10T06:00:00, doni, 62, PTR, GRSK  
2021-11-10T06:00:00, indra, 33, PTR, GRSK  
2021-11-10T06:00:00, indra, 75, SPM, GRSK  
2021-11-10T06:00:00, fakarich, 84, PTR, GRSK  
2021-11-10T06:00:00, doni, 95, SPM, GRSK  
2021-10-28T06:00:00, brian, 34, WLMR, GRSK  
2021-10-28T06:00:00, indra, 99, WLMR, GRSK  
2021-10-28T06:00:00, fakarich, 80, WLMR, GRSK  
2021-11-10T06:00:00, doni, 32, WLMR, GRSK  
2021-11-10T06:00:00, doni, 23, WLMR, GRSK  
2021-11-10T06:00:00, brian, 76, WLMR, GRSK  
2021-11-10T06:00:00, indra, 85, WLMR, GRSK
```

proses penampilan output nama orang yang membeli saham “WLMR_GRSK”

Multiple outputs merupakan proses penampilan lebih dari satu *output* dari suatu hasil proses atau fungsi. Dilakukan pembuatan modul XQuery Transformation bernama “multipleoutputs.xq” dengan *argument* untuk mencari informasi mengenai pembelian saham di “WLMR_GRSK”. Akan ditampilkan semua *output* mengenai informasi pembelian saham, dan *output* nama pembeli (*trader*) saham di “WLMR_GRSK” akan berkumpul menjadi satu dibawah.

7. Accessing Auxiliary Input Data Untuk Mengetahui Jumlah Pembeli Saham Di Tiap Perusahaan



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano accessinput.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar -files trader.txt accessinput.xq -output ./mydata/outputaccessinput -print  
24/03/29 06:17:03 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
```

proses pembuatan direktori modul

```
GNU nano 2.0.9 File: accessinput.xq  
  
import module "oxh:text";  
  
for $visitLine in text:collection("mydata/trading*.log")  
let $visitUserId := fn:tokenize($visitLine, "\s*,\s*")[2]  
  
for $userLine in fn:unparsed-text-lines("trader.txt")  
let $userSplit := fn:tokenize($userLine, "\s*:\s*")  
let $userId := $userSplit[1]  
  
where $userId eq $visitUserId  
  
group by $userId  
return text:put($userId || " " || fn:count($visitLine))
```

proses accessing auxiliary input data

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 06:19:53 INFO hadoop.xquery: Finished executing "accessinput.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputaccessinput"  
brian 7  
doni 12  
fakarich 8  
indra 13  
[oracle@bigdatalite ~]$
```

proses penampilan output nama orang (trader) dan jumlah saham yang dimiliki

Accessing auxiliary input data merupakan proses mengakses data tambahan atau pendukung. Pada proses ini dilakukan sama seperti proses *inner join*, dengan membuat modul XQuery Transformation bernama "accessinput.xq" namun dengan menggunakan data nama *trader* di data trading1.log dan trading2.log kemudian menggunakan nama di trader.txt. Dilakukan proses penampilan *output* yang sama seperti inner joins, yakni penampilan nama orang (*trader*) dan jumlah transaksi pembelian saham yang dilakukan oleh mereka.

8. Calling A Custom Java Function From XQuery (Query Data Input Format Menggunakan Metode Java.Lang.String#format)



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano javafunction.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar javafunction.xq -output ./mydata/outputjavafunction -print  
24/03/29 06:23:03 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
```

proses pembuatan direktori modul

```

oracle@bigdatalite:~
File Edit View Search Terminal Help
GNU nano 2.0.9 File: javafunction.xq

import module "oxh:text";

declare %ora-java:binding("java.lang.String#format")
function local:string-format($pattern as xs:string, $data as xs:anyAtomicType*) as
xs:string external;

for $line in text:collection("mydata/trader*.txt")
let $split := fn:tokenize($line, "\s*:\s*")
return text:put(local:string-format("%s,%s,%s", $split))

proses calling a java function from xquery

oracle@bigdatalite:~
File Edit View Search Terminal Help
24/03/29 06:24:47 INFO hadoop.xquery: Finished executing "javafunction.xq". Output
path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputjavafuncion"
indira,26,Indra Kenz
fakarich,31,Fakar Suhartami
doni,24,Doni Salmanan
brian,27,Brian Edgar Nababan

proses penampilan output setelah ":" diubah menjadi ","

```

Calling a custom java function from XQuery merupakan kemampuan XQuery untuk memanggil atau menggunakan fungsi *custom* yang dibuat dalam pemrograman java yang diambil dari dokumen XML. Di buat modul bernama “javafuncion.xq” dengan menggunakan data trader.txt untuk mengubah “:” menjadi “,”.

9. User Defined XQuery Library Modules And XML Schema

```

oracle@bigdatalite:~/mytools
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ sudo nano mytools.xq
[oracle@bigdatalite ~]$ sudo nano mytools2.xq
[oracle@bigdatalite ~]$ mkdir mytools
[oracle@bigdatalite ~]$ mv mytools.xq mytools2.xq mytools
[oracle@bigdatalite ~]$ cd mytools
[oracle@bigdatalite mytools]$ hadoop jar $OXH_HOME/lib/oxh.jar -files mytools.x
q mytools2.xq -output ./mydata/mydata-out -print
24/03/29 06:36:20 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (buil
d 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.

proses pembuatan direktori mytools.xq dan mytools2.xq

oracle@bigdatalite:~
File Edit View Search Terminal Help
GNU nano 2.0.9 File: mytools.xq Modified

module namespace mytools = "urn:mytools";
declare %ora-java:binding("java.lang.String#format")
function mytools:string-format($pattern as xs:string, $data as xs:anyAtomic$
xs:string external;

proses pembuatan mytools.xq

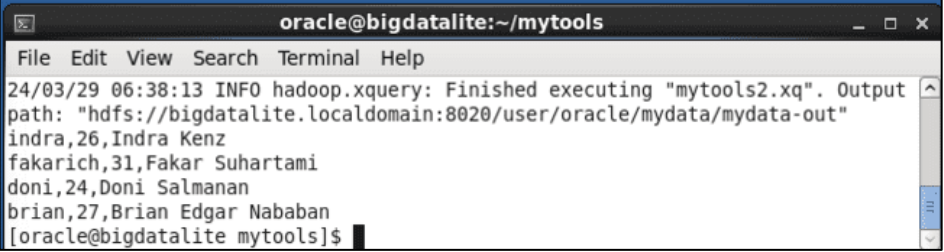
oracle@bigdatalite:~
File Edit View Search Terminal Help
GNU nano 2.0.9 File: mytools2.xq Modified

import module "oxh:text";

for $line in text:collection("mydata/trader*.txt")
let $split := fn:tokenize($line, "\s*:\s*")
return text:put(mytools:string-format("%s,%s,%s", $split))

proses pembuatan mytools2.xq


```

```
oracle@bigdatalite:~/mytools
File Edit View Search Terminal Help
24/03/29 06:38:13 INFO hadoop.xquery: Finished executing "mytools2.xq". Output
path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/mydata-out"
indra,26,Indra Kenz
fakarich,31,Fakar Suhartami
doni,24,Doni Salmanan
brian,27,Brian Edgar Nababan
[oracle@bigdatalite mytools]$
```

proses penampilan output setelah ":" diubah menjadi ","

User defined XQuery library modules and xml schema merupakan percobaan menggunakan sebuah modul *library* dengan pembuatan modul bernama “mytools.xq” untuk memformat *string* dan membuat modul kedua bernama “mytools2.xq” untuk penggunaan modul XQuery Transformation dengan menggunakan *import* modul “mytools.xq”. Dengan membuat direktori “mytools” dilakukan pemindahan modul yang sebelumnya telah dibuat kedalam direktori tersebut. Kemudian, dijalankan direktori “mytools.xq” untuk menampilkan hasil dari proses pengubahan format “:” menjadi “,” menggunakan data trader.txt. Dapat diperlihatkan tampilan dari data trader.txt, trading1.log dan trading2.log, seperti berikut.



```
trader.txt
indra,26,Indra Kenz
fakarich,31,Fakar Suhartami
doni,24,Doni Salmanan
brian,27,Brian Edgar Nababan

trading1.log
2021-10-28T00:00:00,indra,100,SMN_GRSK
2021-10-28T00:00:00,brian,102,PTB_GRSK
2021-10-28T00:00:00,fakarich,98,SMN_GRSK
2021-10-28T00:00:00,doni,86,PTB_GRSK
2021-10-28T00:00:00,indra,108,PTB_GRSK
2021-10-28T00:00:00,brian,45,SMN_GRSK
2021-10-28T00:00:00,indra,106,PTB_GRSK
2021-10-28T00:00:00,doni,77,SMN_GRSK
2021-10-28T00:00:00,doni,67,PTB_GRSK
2021-10-28T00:00:00,fakarich,101,PTB_GRSK
2021-10-28T00:00:00,fakarich,45,MLMR_GRSK
2021-10-28T00:00:00,doni,66,PTB_GRSK
2021-10-28T00:00:00,doni,73,SMN_GRSK
2021-10-28T00:00:00,indra,82,PTB_GRSK
2021-10-28T00:00:00,brian,96,SMN_GRSK
2021-10-28T00:00:00,brian,36,MLMR_GRSK
2021-10-28T00:00:00,indra,109,SMN_GRSK
2021-10-28T00:00:00,indra,99,MLMR_GRSK
2021-10-28T00:00:00,fakarich,118,SMN_GRSK
2021-10-28T00:00:00,fakarich,88,MLMR_GRSK

trading2.log
2021-11-18T00:00:00,brian,64,SMN_GRSK
2021-11-18T00:00:00,fakarich,99,PTB_GRSK
2021-11-18T00:00:00,doni,93,SMN_GRSK
2021-11-18T00:00:00,indra,82,PTB_GRSK
2021-11-18T00:00:00,indra,103,PTB_GRSK
2021-11-18T00:00:00,fakarich,64,SMN_GRSK
2021-11-18T00:00:00,doni,91,PTB_GRSK
2021-11-18T00:00:00,indra,71,SMN_GRSK
2021-11-18T00:00:00,indra,63,PTB_GRSK
2021-11-18T00:00:00,doni,62,PTB_GRSK
2021-11-18T00:00:00,doni,32,MLMR_GRSK
2021-11-18T00:00:00,indra,55,PTB_GRSK
2021-11-18T00:00:00,indra,76,SMN_GRSK
2021-11-18T00:00:00,fakarich,87,PTB_GRSK
2021-11-18T00:00:00,doni,95,SMN_GRSK
2021-11-18T00:00:00,doni,23,MLMR_GRSK
2021-11-18T00:00:00,brian,104,SMN_GRSK
2021-11-18T00:00:00,brian,76,MLMR_GRSK
2021-11-18T00:00:00,doni,119,SMN_GRSK
2021-11-18T00:00:00,indra,85,MLMR_GRSK
```

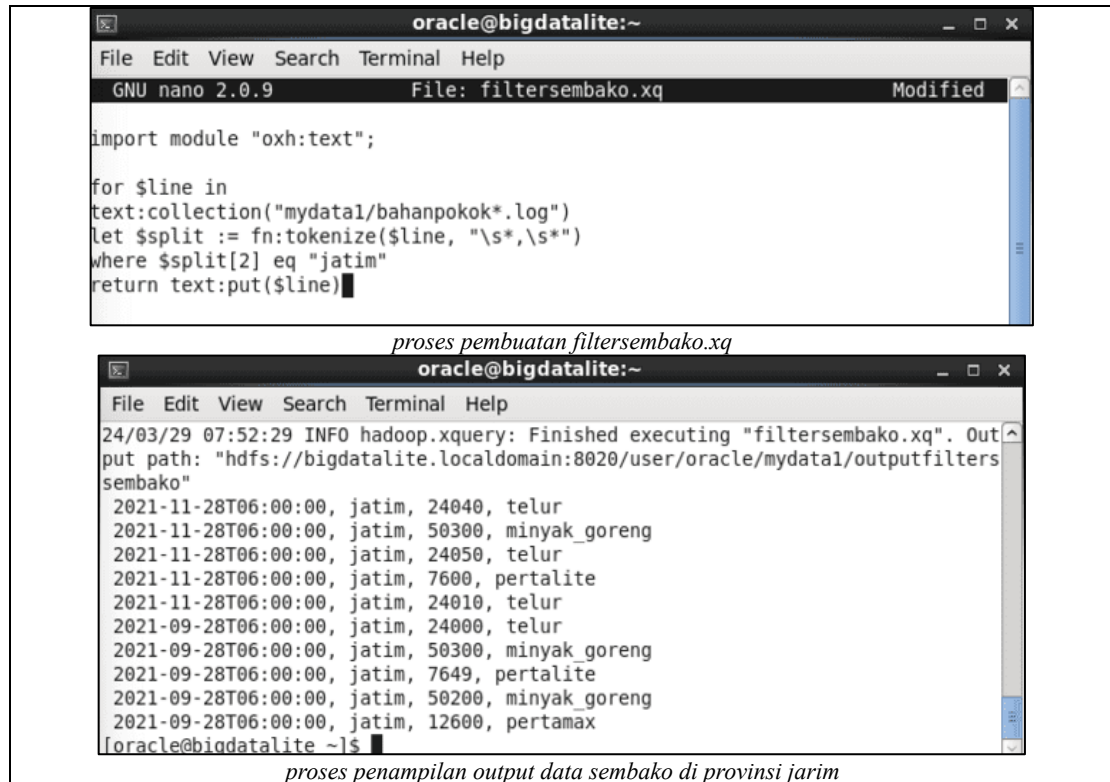
proses penampilan data trader.txt, trading1.log, dan trading2.log

C. Data Kenaikan Harga Sembako

Data kenaikan harga sembako merupakan data yang merepresentasikan catatan atau rekaman dari aktivitas catatan harga sembako ditanggal tertentu dengan masing-masing provinsi. Dimana pada data kenaikan harga sembako ini terjadi menjadi dua, yakni data provinsi.txt yang berisi singkatan nama provinsi dan nama provinsi. Kemudian, terdapat data bahanpokok1.log dan bahanpokok2.log yang berisi tanggal, provinsi, harga, dan nama sembako.

Dengan menggunakan data kenaikan harga sembako akan dilakukan proses penerapan XQuery Transformation yang merupakan konsep pemrosesan data yang mengacu pada ekspresi atau perubahan yang diterapkan pada kueri data. Pada data log trading saham, dilakukan beberapa penerapan XQuery Transformation, meliputi proses *integrating files*, *basic filtering*, *group by and aggregation*, *semi joins*, *accessing auxiliary input data*, dan *custom java function from XQuery*.

1. Integrating File Dan Filtering Data Log Trading Saham Yang Di Beli Brian



Dilakukan penerapan penggabungan file dan filtering dengan menggunakan argument yang ditulis dalam modul “\$sudo nano filtersembako.xq” Dilakukan proses pembuatan direktori bernama “mydata1” dan menyalin file data bahanpokok1.log dan bahanpokok2.log kedalam hadoop. Kemudian, dilakukan proses penggabungan data bahanpokok1.log dan bahanpokok2.log menggunakan argument yang ditulis di dalam modul, yakni “text:collection(“mydata/bahanpokok*.log”)”. Setelah melakukan penggabungan data bahanpokok1.log dan bahanpokok2.log, langkah selanjutnya adalah melakukan proses filtering untuk penampilan informasi mengenai bahan sembako diprovinsi jatim.

Kemudian, dilakukan proses filtering atau pemilihan informasi harga sembako di provinsi “jarim” dengan menggunakan argument yang ditulis didalam modul, yakni “where \$split[2] eq “jatim””. Dan, ditampilkan output tanggal, harga, dan nama sembako yang berada di provinsi jatim (jawa timur).

2. Group By And Aggregation Untuk Menampilkan Tanggal Dan Jumlah Kemunculan Sembako Pada Setiap Tanggal



The first screenshot shows the creation of a module named `groupbyaggregation1.xq` in the `nano` editor. The code imports the `oxh:text` module and processes a collection of log files (`mydata1/bahanpokok*.log`) by tokenizing each line, extracting a date, grouping by that date, and returning the count of lines for each date.

The second screenshot shows the execution of the module. The terminal output indicates that the module was executed successfully, and the output path is `hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputgroupbyaggregation1`. The output shows two dates: `2021-09-28 => 20` and `2021-11-28 => 20`.

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: groupbyaggregation1.xq  
  
import module "oxh:text";  
  
for $line in text:collection("mydata1/bahanpokok*.log")  
let $split := fn:tokenize($line, "\s*,\s*")  
let $time := xs:dateTime($split[1])  
let $day := xs:date($time)  
group by $day  
return text:put($day || " => " || fn:count($line))  
  
proses pembuatan modul groupbyaggregation
```

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 08:02:22 INFO hadoop.xquery: Finished executing "groupbyaggregation1.xq"  
". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputgroupbyaggregation1"  
2021-09-28 => 20  
2021-11-28 => 20  
[oracle@bigdatalite ~]$  
  
proses penampilan output tanggal dan jumlah sembako di setiap tanggal
```

Dilakukan penerapan *group by aggregation* untuk menampilkan tanggal dan jumlah saham yang dibeli berdasarkan data pada `bahanpokok1.log` dan `bahanpokok2.log`. Dengan melakukan pembuatan modul yang bernama "`groupbyaggregation1.xq`" dengan *argument* menggunakan semua data `bahanpokok` dan melakukan pengambilan data kolom pertama yang diubah formatnya menjadi tanggal. Kemudian, dilakukan pengelompokkan berdasarkan tanggal, dan ditampilkan *output* pada tanggal 28 bulan 09 tahun 2021 dilakukan penampilan sembako sebanyak 20 serta pada tanggal 28 bulan 11 tahun 2021 dilakukan penambilan sembako sebanyak 20.

3. Semi Joins Untuk Mengetahui Provinsi Yang Memiliki Data Informasi Harga Sembako

Semi join merupakan jenis penggabungan (*join*) yang menghasilkan kumpulan data dari tabel kiri yang memiliki nilai sesuai ditabel kanan. Perlu diketahui bahwa sebelumnya telah melakukan penggabungan data `bahanpokok1.log` dan `bahanpokok2.log`. Kemudian, akan dilakukan proses penyalinan data "`provinsi.txt`" ke dalam `hadoop` dan membuat modul "`datasembako.xq`" untuk menampilkan *output* dari `sembako.txt` dengan penambahan kalimat "`in province`" disetiap akhir baris datanya.

The first screenshot shows the creation of a module named `datasembako.xq` in the `nano` editor. The code imports the `oxh:text` module and processes a collection of province data (`provinsi*.txt`) by returning each line with the text "`in province`" appended to it.

The second screenshot shows the execution of the module. The terminal output indicates that the module was executed successfully, and the output path is `hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputdatasembako`.

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano datasembako.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar datasembako.xq -output  
./myout1b -print  
24/03/29 07:13:55 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build  
4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.  
  
proses pembuatan direktori modul
```

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: datasembako.xq  
  
import module "oxh:text";  
for $line in text:collection ("provinsi*.txt")  
return text:put($line || ",in province")
```

proses datasembako.xq

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 07:15:58 INFO hadoop.xquery: Finished executing "datasembako.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/myout1b"  
sumut::Sumatera Utara,in province  
sumbar::Sumatera Barat,in province  
jabar::Jawa Barat,in province  
jatim::Jawa Timur,in province  
jateng::Jawa Tengah,in province  
[oracle@bigdatalite ~]$
```

proses penampilan nama provinsi

Dilakukan pembuatan modul XQuery Transformation bernama “semijoin1.xq” dengan *argument* menggunakan kolom pertama ditabel provinsi.txt yang berisi nama provinsi dan kolom kedua dari data tabel bahanpokok1.log dan bahanpokok2.log yang berisi nama provinsi yang memiliki tanggal dan informasi harga sembako. Ditampilkan hasil, jabar, jateng, jatim, sumbar, dan sumut memiliki data informasi harga sembako.

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano semijoin1.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar semijoin1.xq -output ./mydata1/outputsemijoins1 -print  
24/03/29 11:13:43 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
```

proses pembuatan direktori modul

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: semijoin1.xq Modified  
import module "oxh:text";  
  
for $userLine in text:collection("mydata1/provinsi.txt")  
let $userId := fn:tokenize($userLine, "\s*\s*")[1]  
  
where some $visitLine in text:collection("mydata1/bahanpokok*.log")  
satisfies $userId eq fn:tokenize($visitLine, "\s*\s*")[2]  
return text:put($userId)
```

proses semijoin1.xq

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 11:23:34 INFO hadoop.xquery: Finished executing "semijoin1.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata1/outputsemijoins1"  
jabar  
jateng  
jatim  
sumbar  
sumut  
[oracle@bigdatalite ~]$
```

proses penampilan nama provinsi dalam "provinsi.txt" yang sesuai dengan "bahanpokok.log"*

4. Accessing Auxiliary Input Data Untuk Mengetahui Jumlah Kenaikan Harga Sembako Yang Di Miliki Tiap Provinsi

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano accessingaux1.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar -files provinsi.txt acc  
essingaux1.xq -output ./mydata/outputaccesingaux1 -print  
24/03/29 11:31:25 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build  
4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
```

proses pembuatan direktori modul

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: accessingaux1.xq  
import module "oxh:text";  
  
for $visitLine in text:collection("mydata/bahanpokok*.log")  
let $visitUserId := fn:tokenize($visitLine, "\s*,\s*")[2]  
  
for $userLine in fn:unparsed-text-lines("provinsi.txt")  
let $userSplit := fn:tokenize($userLine, "\s*:\s*")  
let $userId := $userSplit[1]  
  
where $userId eq $visitUserId  
  
group by $userId  
return text:put($userId || " " || fn:count($visitLine))
```

proses accesing auxiliary input data

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 11:35:28 INFO hadoop.xquery: Finished executing "accessingaux1.xq". Out  
put path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/outputaccesin  
gaux1"  
jabar 7  
jateng 6  
jatim 10  
sumbar 8  
sumut 8  
[oracle@bigdatalite ~]$
```

proses penampilan output provinsi dan data sembako yang dimiliki

Accesing auxiliary input data merupakan proses mengakses data tambahan atau pendukung. Pada proses ini dilakukan sama seperti proses *inner join*, dengan membuat modul XQuery Transformation bernama "acesinput1.xq" namun dengan menggunakan data nama provinsi di data bahanpokok1.log dan bahanpokok2.log kemudian menggunakan nama provinsi di bahanpokok.txt. Dilakukan proses penampilan *output* yang sama seperti inner joins, yakni penampilan nama provinsi dan jumlah harga sembako yang dimilikinya. Dengan *output* jabar sebanyak 7, jateng sebanyak 6, jatim sebanyak 10, sumbar sebanyak 8, dan sumur sebanyak 8.

5. Calling A Custom Java Function From XQuery (Query Data Input Format Menggunakan Metode Java.Lang.String#format)

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ sudo nano javafunction1.xq  
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar javafunction1.xq -outpu  
t ./mydata/outputjavafunction1 -print  
24/03/29 11:41:39 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build  
4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2024, Oracle. All rights reserved.
```

proses pembuatan direktori modul

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: javafunction1.xq Modified  
  
import module "oxh:text";  
  
declare %ora-java:binding("java.lang.String#format")  
function local:string-format($pattern as xs:string, $data as xs:anyAtomicType*)$  
xs:string external;  
  
for $line in text:collection("mydata1/provinsi*.txt")  
let $split := fn:tokenize($line, "\s*:\s*")  
return text:put(local:string-format("%s,%s,%s", $split))
```

proses calling a java function from xquery

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
24/03/29 11:43:24 INFO hadoop.xquery: Finished executing "javafunction1.xq". Out  
put path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata1/outputjavafun  
ction1"  
sumut,,Sumatera Utara  
sumbar,,Sumatera Barat  
jabar,,Jawa Barat  
jatim,,Jawa Timur  
jateng,,Jawa Tengah  
[oracle@bigdatalite ~]$
```

proses penampilan output setelah ":" diubah menjadi ","

Calling a custom java function from XQuery merupakan kemampuan XQuery untuk memanggil atau menggunakan fungsi *custom* yang dibuat dalam pemrograman java yang diambil dari dokumen XML. Di buat modul bernama "javafunction1.xq" dengan menggunakan data provinsi.txt untuk mengubah ":" menjadi ",".

Dapat diperlihatkan tampilan dari data provinsitxt, bahanpokok1.log dan bahanpokok2.log, seperti berikut.

provinsi.txt	bahanpokok1.log	bahanpokok2.log
sumut::Sumatera Utara sumbar::Sumatera Barat jabar::Jawa Barat jatim::Jawa Timur jateng::Jawa Tengah	2021-09-28T06:00:00, jatim, 24000, telur 2021-09-28T06:00:00, jateng, 58000, minyak_goreng 2021-09-28T06:00:00, jakar, 7642, pertalite 2021-09-28T06:00:00, sumbar, 12500, pertamax 2021-09-28T06:00:00, sumut, 7633, pertalite 2021-09-28T06:00:00, sumbar, 12400, pertamax 2021-09-28T06:00:00, jateng, 24000, telur 2021-09-28T06:00:00, jatim, 58300, minyak_goreng 2021-09-28T06:00:00, jatim, 7640, pertalite 2021-09-28T06:00:00, jateng, 12400, pertamax 2021-09-28T06:00:00, jakar, 23400, telur 2021-09-28T06:00:00, jatim, 58200, minyak_goreng 2021-09-28T06:00:00, jakar, 7630, pertalite 2021-09-28T06:00:00, jakar, 58300, minyak_goreng 2021-09-28T06:00:00, jakar, 12520, pertamax 2021-09-28T06:00:00, jakar, 24020, telur 2021-09-28T06:00:00, sumbar, 58300, minyak_goreng 2021-09-28T06:00:00, sumut, 7630, pertalite 2021-09-28T06:00:00, jatim, 12000, pertamax 2021-09-28T06:00:00, sumut, 24000, telur 2021-09-28T06:00:00, sumbar, 58200, minyak_goreng	2021-11-28T06:00:00, jateng, 58000, minyak_goreng 2021-11-28T06:00:00, pertalite, 7632, pertalite 2021-11-28T06:00:00, jakar, 7642, pertalite 2021-11-28T06:00:00, sumut, 2405, telur 2021-11-28T06:00:00, sumut, 58400, minyak_goreng 2021-11-28T06:00:00, sumbar, 7630, pertalite 2021-11-28T06:00:00, jatim, 24000, telur 2021-11-28T06:00:00, jatim, 58300, minyak_goreng 2021-11-28T06:00:00, jatim, 24050, telur 2021-11-28T06:00:00, jateng, 58300, minyak_goreng 2021-11-28T06:00:00, jakar, 23430, telur 2021-11-28T06:00:00, jateng, 58100, minyak_goreng 2021-11-28T06:00:00, jatim, 7600, pertalite 2021-11-28T06:00:00, sumut, 12510, pertamax 2021-11-28T06:00:00, jatim, 24010, telur 2021-11-28T06:00:00, sumbar, 58300, minyak_goreng 2021-11-28T06:00:00, sumut, 7630, pertalite 2021-11-28T06:00:00, sumbar, 12500, pertamax 2021-11-28T06:00:00, sumut, 24000, telur 2021-11-28T06:00:00, sumut, 58100, minyak_goreng

proses penampilan data provinsi.txt, bahanpokok1.log, dan bahanpokok2.log

Kesimpulan :

Dalam penugasan ini, dilakukan analisis pada studi kasus menggunakan 3 dataset, yakni data Noaa, data log trading saham, dan data kenaikan sembako. Pada data NOAA yang terdiri dari tahun 1901 hingga 1930 dilakukan pemrosesan query statistik deskriptif seperti mencari nilai maksimum, minimum, rata-rata, varian, standar deviasi, korelasi antar variabel, analisis kenaikan suhu, dan deteksi nilai ekstrem suhu di tiap tahun.

Kemudian, dilakukan penerapan XQuery Transformation pada data log trading saham dan data harga sembako dengan menerapkan beberapa proses, seperti *integrating files, basic filtering, group by and aggregation, inner joins, left outer joins, semi joins, multiple outputs, accesing auxiliary input data, custom java function from XQuery*, dan *user-defined XQuery library modules and XML schemas*. Link dokumen yang belum di compress : https://drive.google.com/drive/folders/1sW_XQ80VkSfrGotPEIVzoPROX-4h6aNn?usp=drive_link