LAPORAN
RENCANA TUGAS MANDIRI (RTM) Ke-IV
MATA KULIAH BIG DATA KELAS B

DISUSUN OLEH:

Aulia Nur Fitriani (21083010051)

DOSEN PENGAMPU:

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

PROGRAM STUDI SAINS DATA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS PEMBANGUNAN NASIONAL "VETERAN"

JAWA TIMUR

2023

Rencana Tugas-4: Query statistik deskriptif menggunakan Hive dan XQuery

1. Silakan lakukan analisis pada dataset NOAA menggunakan Hive untuk menjawab pertanyaan:

a. Statistika deskriptif (suhu maksimum, minimum, rata-rata, varian, deviasi standar, dan persentil) yang dikelompokkkan berdasarkan masing-masing tahun.

b. Persentase perubahan rata-rata suhu di antara 2 tahun, misalnya antara tahun 1902-1903

Selanjutnya, buatlah 3 pertanyaan tambahan analisis berdasarkan dataset NOAA tersebut (3 kolom) dan jawablah menggunakan sintaks query serta tampilkan hasilnya

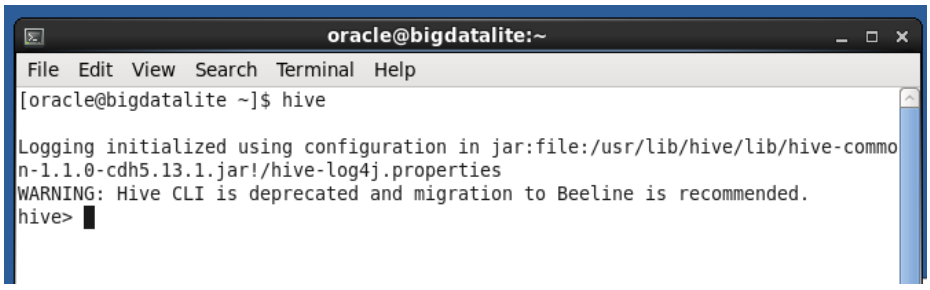
2. Unduh dataset dummy - Saham dan Harga Sembako (kerjakan keduanya) yang dapat diakses di https://drive.google.com/drive/folders/182b5TikHcqCe2vAzgfabaNNjAfG5Qh6s?usp=sharing. Lalu analisislah menggunakan bentuk-bentuk Xquery transformation yang sesuai

JAWABAN

1. Silakan lakukan analisis pada dataset NOAA menggunakan Hive untuk menjawab pertanyaan:

a) Statistika deskriptif (suhu maksimum, minimum, rata-rata, varian, deviasi standar, dan persentil) yang dikelompokkkan berdasarkan masing-masing tahun.

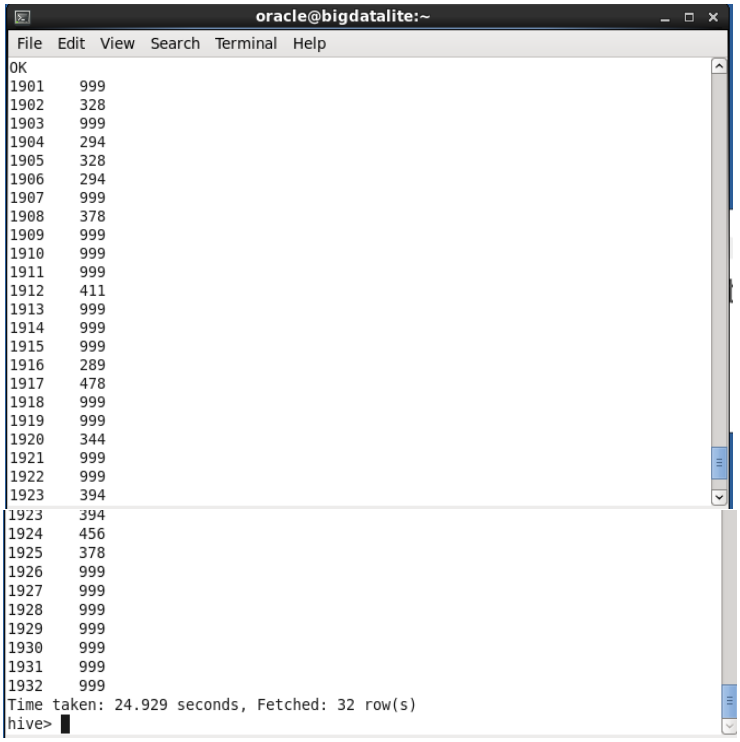
Sebelum menghitung nilai-niai nya kita harus masuk dulu kedalam hive seperti gambar berikut :



- **Suhu Maksimum**
  Untuk menghitung nilai suhu maksimum nya menggunakan syntax berikut :

```
SELECT tahun, MAX(suhu) AS suhu_max
FROM suhutemp
GROUP BY tahun;
```

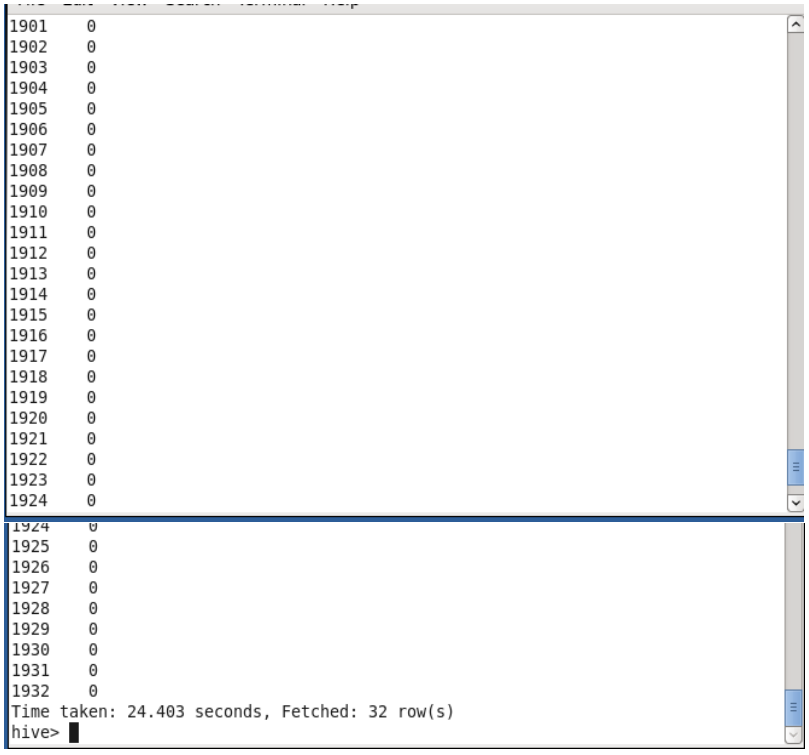
Output :



- **Suhu Minimum**
  Untuk menghitung nilai suhu minimum nya menggunakan syntax berikut :

```
SELECT tahun, MIN(suhu) AS suhu_min
FROM suhutemp
GROUP BY tahun;
```

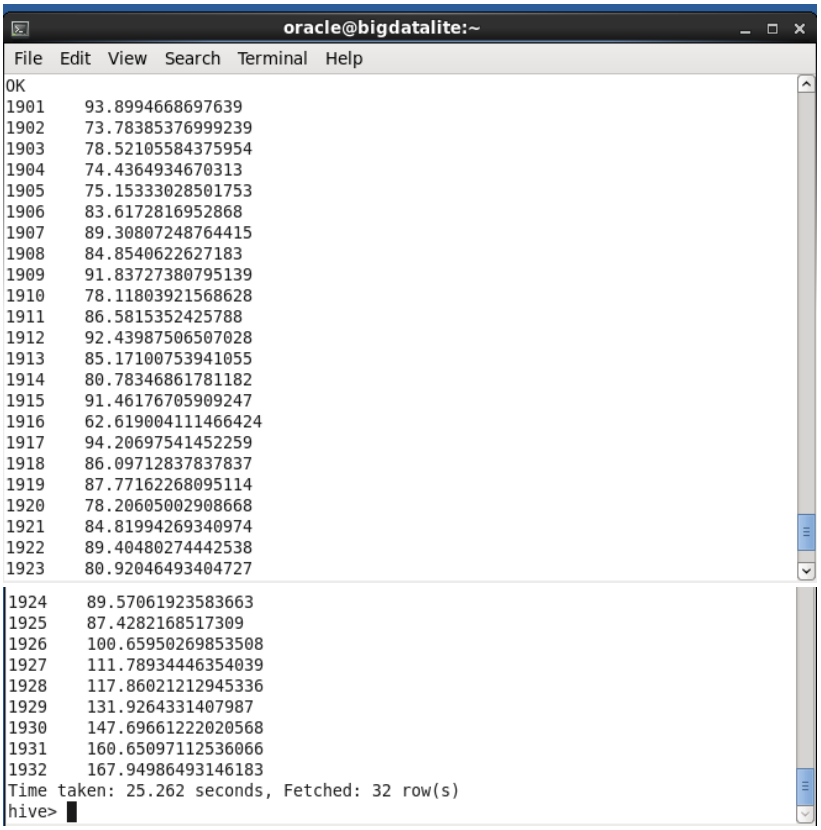
Output:



- **Rata-Rata**
  Untuk menghitung nilai rata-rata nya menggunakan syntax berikut :

```
SELECT tahun, AVG(suhu) AS suhu_rata
FROM suhutemp
GROUP BY tahun;
```

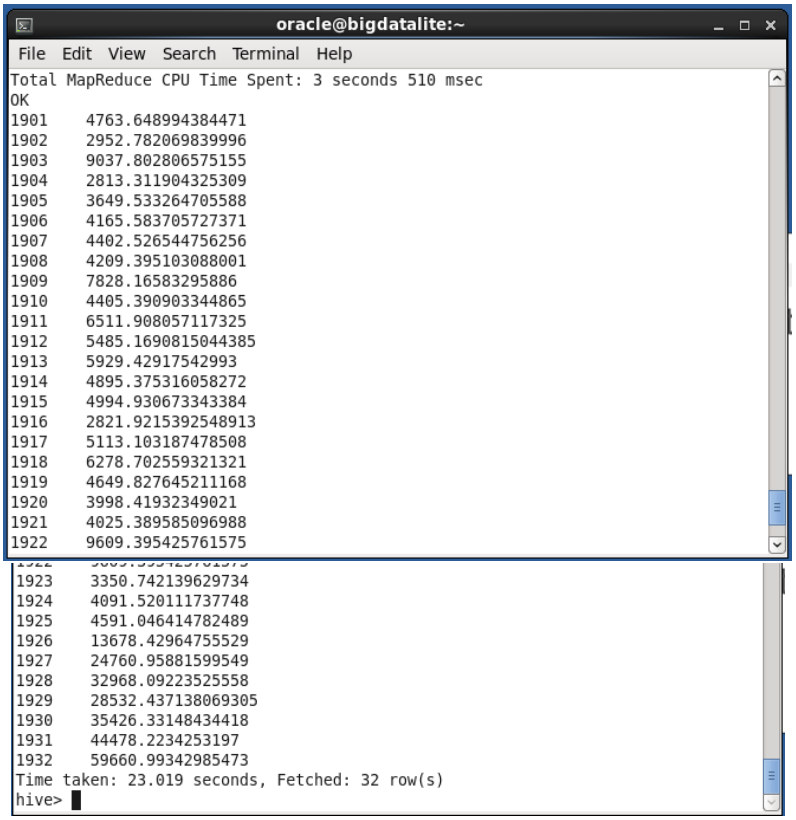
Output:



- **Varian**
  Untuk menghitung nilai varian nya menggunakan syntax berikut :

```
SELECT tahun, VAR_POP(suhu) AS varian_suhu
FROM suhutemp
GROUP BY tahun;
```

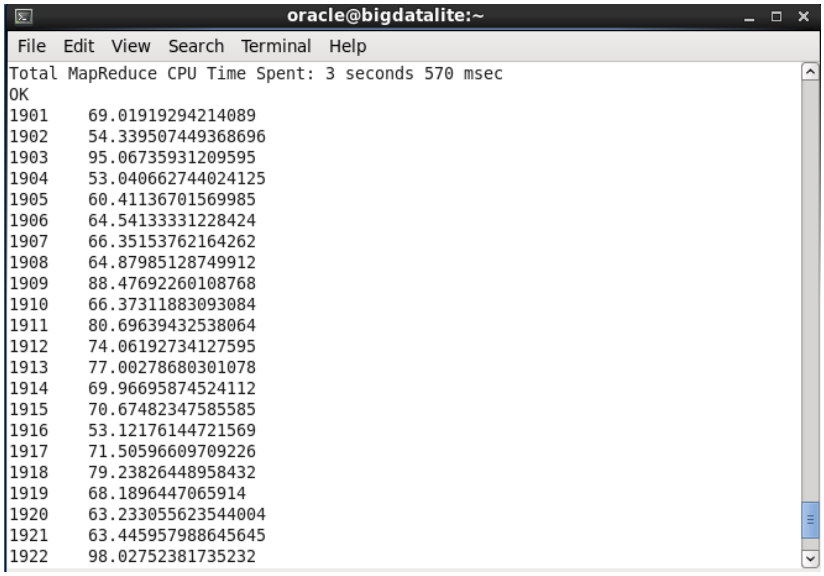
Output:



- **Standar Deviasi**
  Untuk menghitung nilai standar deviasi nya menggunakan syntax berikut :

```
SELECT tahun, STDDEV_POP(suhu) AS
deviasi_standar_suhu
FROM suhutemp
GROUP BY tahun;
```

Output:



```
1922    98.02752381753252
1923    57.8855952688554
1924    63.96499129787909
1925    67.75726097461798
1926    116.95481882998789
1927    157.35615277451177
1928    181.57117677444177
1929    168.9154733530037
1930    188.2188393449077
1931    210.89860934894688
1932    244.25599978271717
Time taken: 26.935 seconds, Fetched: 32 row(s)
hive>
```

- **Persentil**
  Untuk menghitung nilai persentil nya menggunakan syntax berikut :

```
SELECT tahun, percentile(suhu, 0.25) AS suhu_p25,
percentile(suhu, 0.5) AS suhu_p50,
percentile(suhu, 0.75) AS suhu_p75
FROM suhutemp
GROUP BY tahun;
```

  Output:

```
oracle@bigdatalite:~
File  Edit  View  Search  Terminal  Help
OK
1901    33.0    89.0    144.0
1902    28.0    67.0    111.0
1903    22.0    56.0    122.0
1904    28.0    67.0    117.0
1905    22.0    61.0    122.0
1906    28.0    72.0    128.0
1907    33.0    83.0    133.0
1908    28.0    72.0    128.0
1909    33.0    83.0    133.0
1910    22.0    67.0    122.0
1911    28.0    72.0    128.0
1912    28.0    78.0    144.0
1913    28.0    72.0    128.0
1914    28.0    61.0    117.0
1915    33.0    78.0    133.0
1916    22.0    50.0    89.0
1917    33.0    78.0    144.0
1918    33.0    72.0    122.0
1919    28.0    78.0    128.0
1920    22.0    61.0    122.0
1921    28.0    78.0    128.0
1922    28.0    72.0    128.0
1923    33.0    72.0    122.0
1924    33.0    83.0    133.0
1925    28.0    78.0    133.0
1926    39.0    78.0    139.0
1927    39.0    78.0    128.0
1928    39.0    78.0    128.0
1929    61.0    100.0   144.0
1930    61.0    111.0   156.0
1931    56.0    111.0   178.0
1932    50.0    100.0   161.0
Time taken: 30.38 seconds, Fetched: 32 row(s)
hive>
```

b) Persentase perubahan rata-rata suhu di antara 2 tahun, misalnya antara tahun 1902-1903

```
SELECT (rata_rata_1903 - rata_rata_1902) /
rata_rata_1902 * 100 AS
persentase_perubahan_rata_rata_suhu
FROM (SELECT AVG(suhu) AS rata_rata_1902
      FROM suhutemp
      WHERE tahun = 1902) t1
JOIN (SELECT AVG(suhu) AS rata_rata_1903
      FROM suhutemp
      WHERE tahun = 1903) t2;
```

Output:

```
OK
6.420377673053657
Time taken: 92.486 seconds, Fetched: 1 row(s)
hive>
```

Selanjutnya yaitu membuat 3 pertanyaan :

1. Berapa persentase kenaikan suhu rata-rata antara tahun 1920 dan 1921?
   Untuk menghitung nilai nya menggunakan syntax berikut :

```
SELECT ((t2.rata_rata - t1.rata_rata) / t1.rata_rata) *
100 AS persentase_perubahan_rata_rata_suhu
FROM (SELECT AVG(suhu) AS rata_rata
      FROM suhutemp
      WHERE tahun = 1920) t1
JOIN (SELECT AVG(suhu) AS rata_rata
      FROM suhutemp
      WHERE tahun = 1921) t2;
```

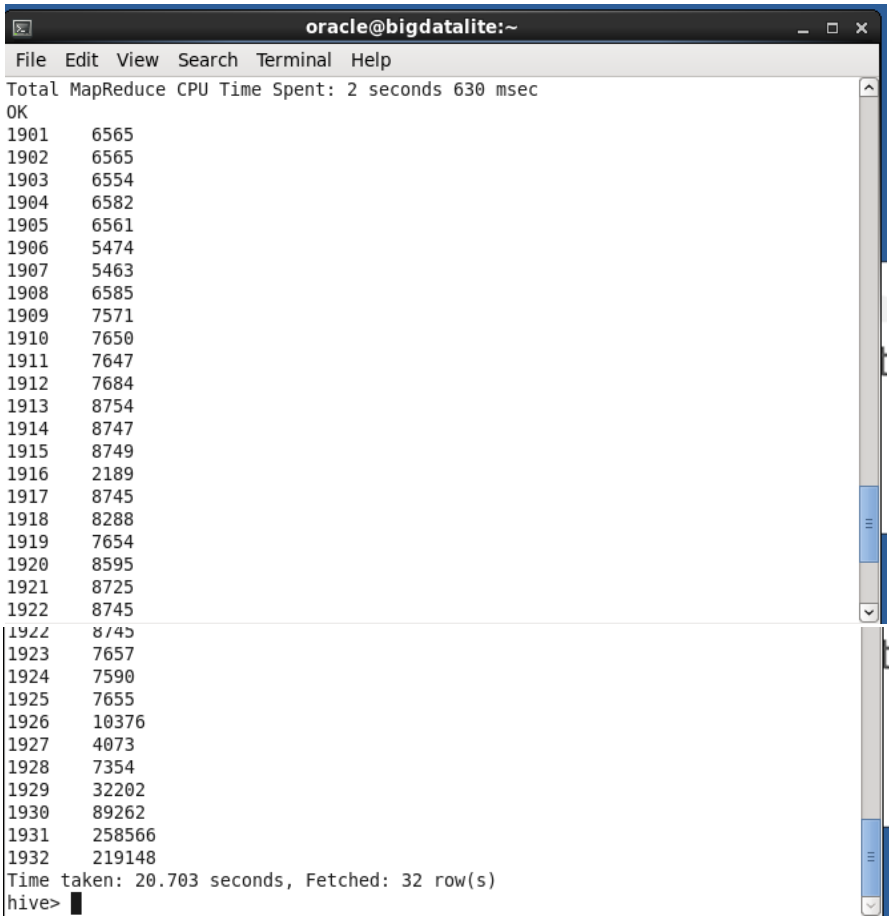
   Output :

```
8 SUCCESS
Total MapReduce CPU Time Spent: 8 seconds 580 msec
OK
8.457008967801336
Time taken: 64.595 seconds, Fetched: 1 row(s)
hive>
```

2. Berapa banyak data suhu yang diambil pada setiap tahun?
   Untuk menghitung nilai nya menggunakan syntax berikut :

```
SELECT tahun, COUNT(*) AS jumlah_data
FROM suhutemp
GROUP BY tahun;
```

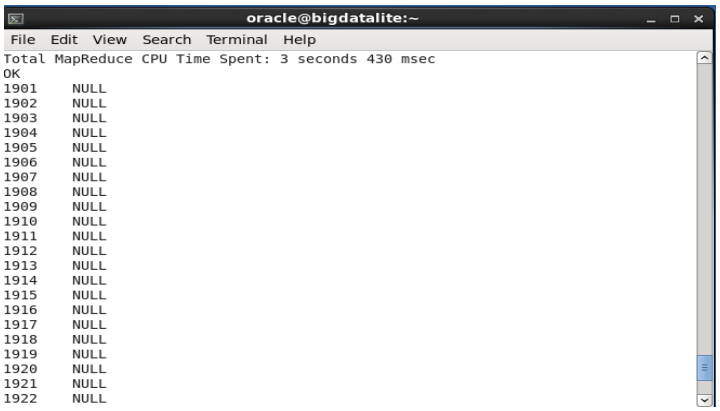
   Output:



3. Apakah ada korelasi antara suhu maksimum dan suhu minimum pada setiap tahun? Untuk menghitung nilai nya menggunakan syntax berikut :

```
SELECT tahun, CORR(suhu_maksimum, suhu_minimum) AS
korelasi_suhu
FROM (SELECT tahun, MAX(suhu) AS suhu_maksimum,
MIN(suhu) AS suhu_minimum
      FROM suhutemp
      WHERE suhu IS NOT NULL
      GROUP BY tahun) t
GROUP BY tahun;
```

Output:



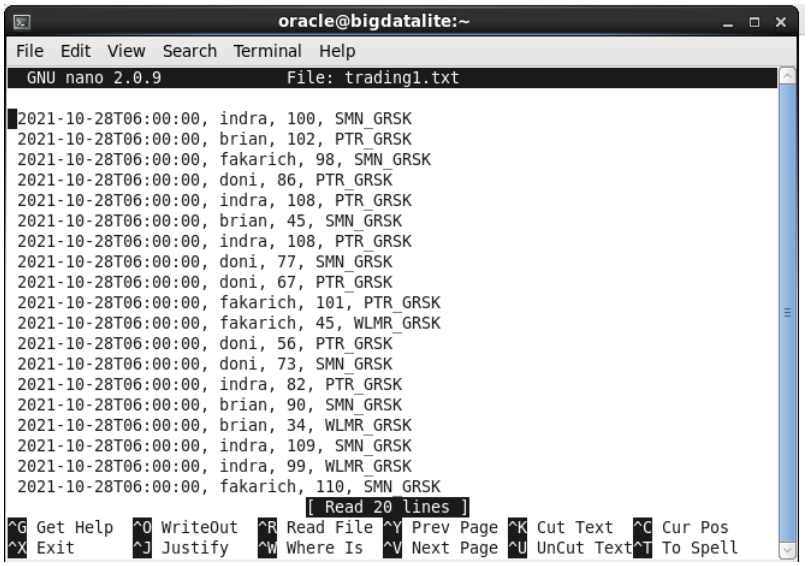
Nilai null pada data tidak akan mempengaruhi hasil korelasi antara suhu maksimum dan suhu minimum pada setiap tahun. Jadi tidak ada korelasi antara suhu maksimum dan suhu minimum

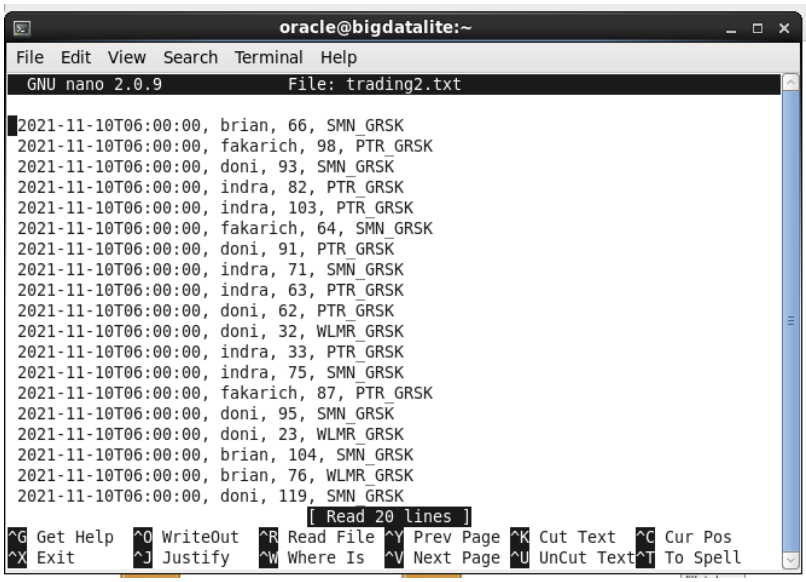
2. Analisis menggunakan bentuk-bentuk Xquery transformation

Ganjil

- **Xquery Sederhana**

1. Mengubah file trading1.log dan trading2.log kedalam file .txt
2. Membuka file dengan perintah nano kemudian menyimpannya

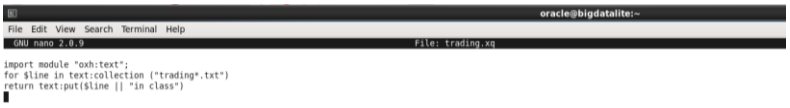




3. Simpan file kedalam hdfs

```
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal trading1.txt
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal trading2.txt
```

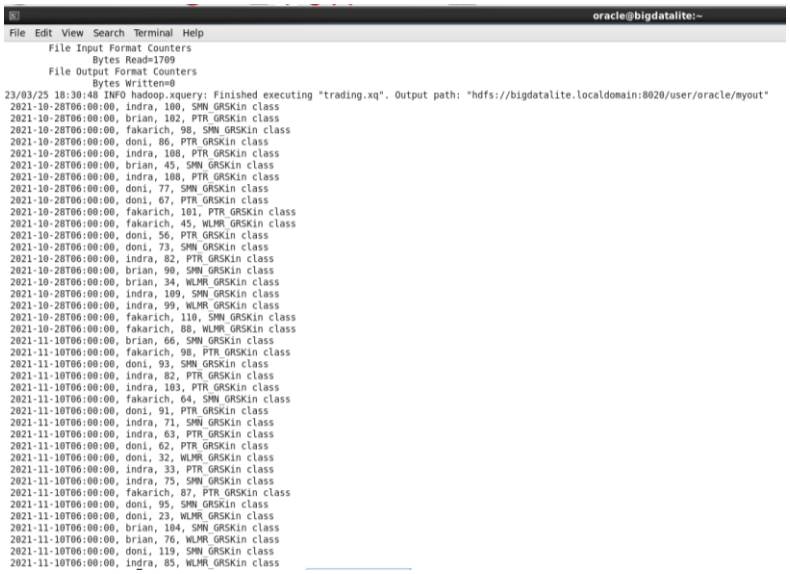
4. Menggabungkan kedua file trading kedalam file .xq



5. Untuk menampilkan file trading.xq ke hadoop dengan pentintah berikut ini

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar trading.xq -output ./myout -print
```

Hasilnya :

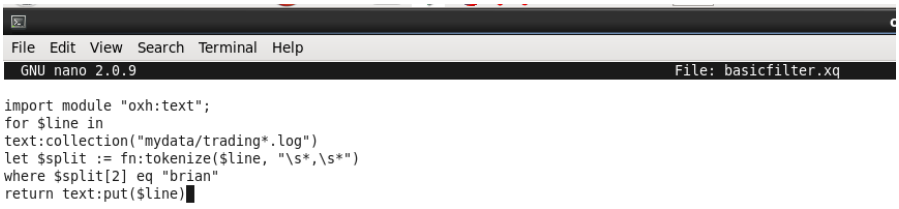


- **XQuery Basic Filtering**
  1. Buat direktori baru pada hadoop filesystem dengan nama “/mydata”

     ```
     [oracle@bigdatalite ~]$ hadoop fs -mkdir -p /user/oracle/mydata
     ```
  2. Copy file trading1.log dan trading2.log yang telah dibuat

     ```
     [oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal trading1.log /user/oracle/mydata
     [oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal trading2.log /user/oracle/mydata
     ```

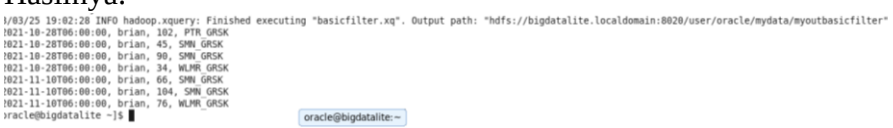
  3. Membuat file baru dengan nama “basicfilter.xq” untuk melakukan query dengan menggunakan filter halaman yang dikunjungi oleh user bernama ‘brian’

  4. Menjalankan Xquery diatas dengan syntax

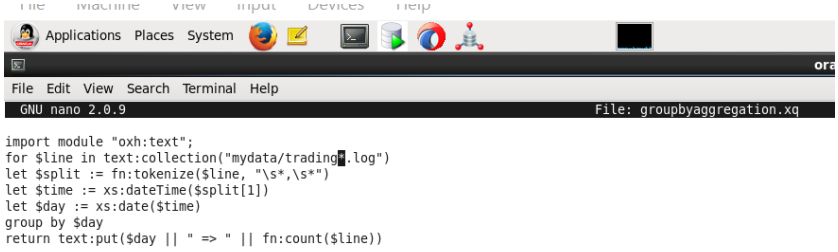
     ```
     [oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar basicfilter.xq -output./mydata/myoutbasicfilter -print
     ```

     Hasilnya:

- **Group by and Aggregation**
  1. Membuat file Xquery dengan nama “groupbyaggregation.xq” XQuery tersebut berguna untuk mengetahui berapa banyak kunjungan user tiap hari

berdasarkan per tanggal



2. Menjalankan Xquery diatas dengan syntax

```
oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar groupbyaggregation.xq -output ./mydata/myoutgroupbyaggregation -print
```

Hasilnya:

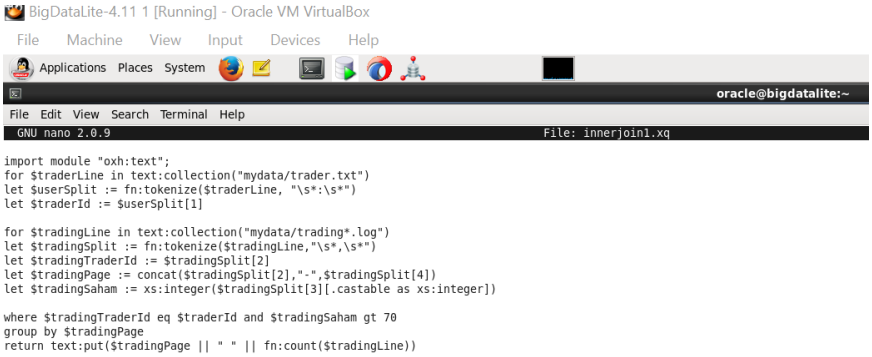
```
23/03/25 19:10:18 INFO hadoop.xquery: Finished executing "grou
2021-10-28 => 20
2021-11-10 => 20
[oracle@bigdatalite ~]$
```

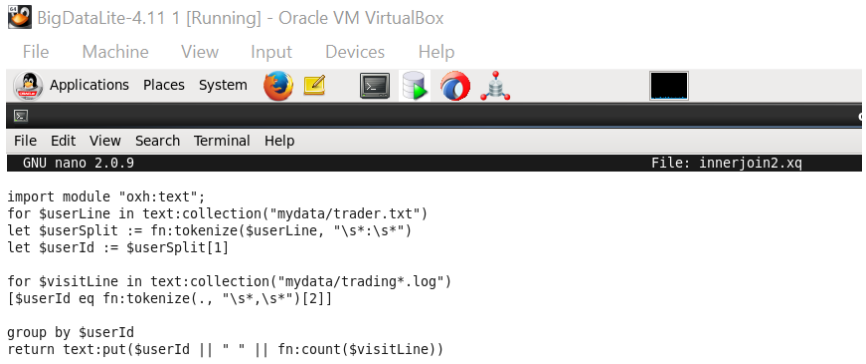
- **Inner Joins**

1. Copy file trader.txt ke dalam directory hadoop

```
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal trader.txt /user/oracle/mydata
[oracle@bigdatalite ~]$
```

2. Membuat file Xquery dengan nama innerjoin1.xq dan innerjoin2.xq





3. Menjalankan Xquery diatas dengan syntax

Innerjoin1.xq

```
racle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar innerjoin1.xq -output ./mydata/myoutinnerjoin1 -print
```

Innerjoin2.xq

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar innerjoin2.xq -output ./mydata/myoutinnerjoin2 -print
```

Hasilnya innerjoin1.xq:

```
23/03/25 19:28:43 INFO hadoop.xquery: Finished executing "innerjoin1.xq". Ou
brian-PTR_GRSK 1
brian-SMN_GRSK 2
brian-WLMR_GRSK 1
doni-PTR_GRSK 2
doni-SMN_GRSK 5
fakarich-PTR_GRSK 3
fakarich-SMN_GRSK 2
fakarich-WLMR_GRSK 1
indra-PTR_GRSK 5
indra-SMN_GRSK 4
indra-WLMR_GRSK 2
[oracle@bigdatalite ~]$
```

oracle@bigdatalite:~

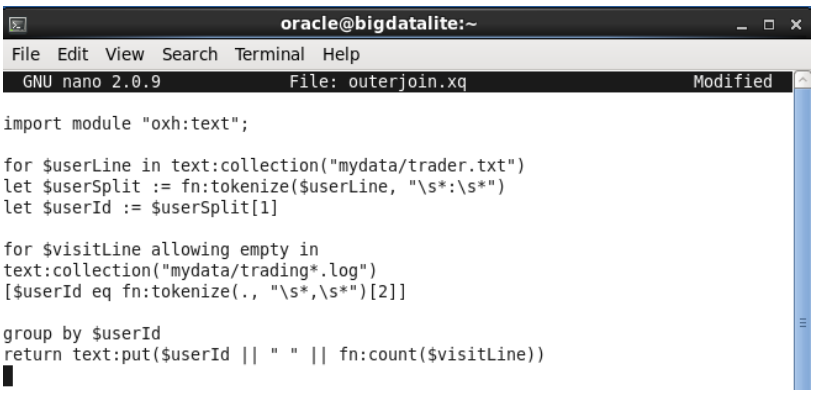
Hasilnya innerjoin2.xq:

```
23/03/25 19:34:28 INFO hadoop.xquery: Finished executing "innerjoin2.xq". Outp
brian 7
doni 12
fakarich 8
indra 13
[oracle@bigdatalite ~]$
```

oracle@bigdatalite:~

- **Left Outer Joins**

1. Membuat Xquery dengan nama “outerjoin.xq” XQuery tersebut berguna untuk menampilkan dna menghitung user yang mengakses suatu halaman maupun yang tidak dengan tambahan sintaks query ‘allowing empty’.



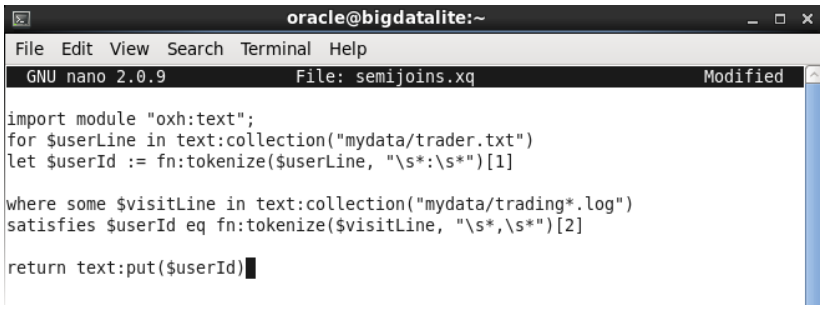
2. Menjalankan Xquery diatas dengan syntax

```
oracle@bigdatalite ~]$ sudo nano outerjoin.xq
oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar outerjoin.xq -output ./
ydata/myoutouterjoin -print
```

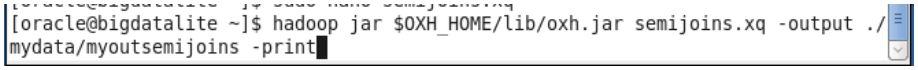
Hasilnya :

```
                Bytes Written=0
23/03/25 19:44:04 INFO hadoop.xquery: Finished executing "outerjoin.xq". Output
path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/myoutouterjoin"
brian 7
doni 12
fakarich 8
indra 13
[oracle@bigdatalite ~]$
```

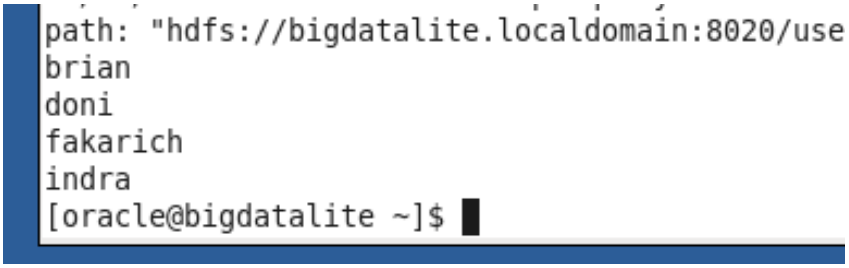
- **Semijoins**
  1. Membuat Xquery dengan nama “semijoins.xq”
     XQuery tersebut berguna untuk menampilkan user siapa saja yang pernah mengakses
     halaman namun secara sintaks query menggunakan “where” condition dan “satisfies”
     berdasarkan kecocokan antara id user dan data kunjungan user.
     Sehingga user yang tidak pernah mengunjungi halaman apapun, tidak akan ditampilkan,
     dikarenakan tidak ada id user yang cocok dengan data kunjungan user yang bersangkutan.



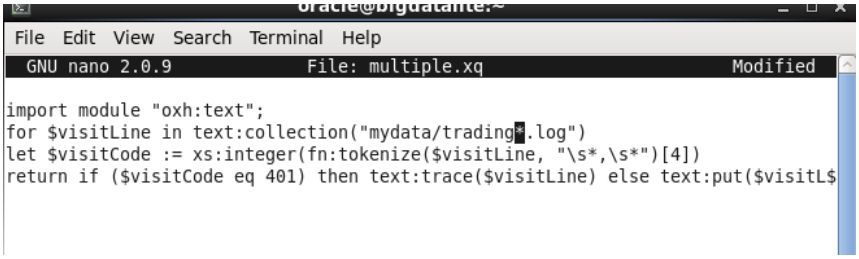
  2. Menjalankan Xquery diatas dengan syntax



     Hasilnya :



- **Multiple Outputs**
  1. Membuat Xquery dengan nama “multipleoutputs.xq”



  2. Menjalankan Xquery diatas dengan syntax

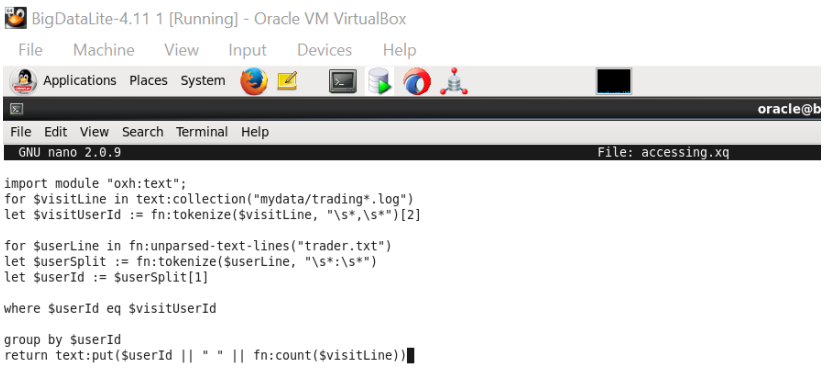
```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar multiple.xq -output ./m
ydata/myoutmultiple -print
```

Hasilnya :

```
23/03/25 13:40:42 INFO hadoop.xquery: Finished executing "multiple.xq"
atalite.localdomain:8020/user/oracle/mydata/myoutmultiple"
 2021-10-28T06:00:00, indra, 100, SMN_GRSK
 2021-10-28T06:00:00, brian, 102, PTR_GRSK
 2021-10-28T06:00:00, fakarich, 98, SMN_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, SMN_GRSK
 2021-10-28T06:00:00, indra, 108, PTR_GRSK
 2021-10-28T06:00:00, doni, 77, SMN_GRSK
 2021-10-28T06:00:00, doni, 67, PTR_GRSK
 2021-10-28T06:00:00, fakarich, 101, PTR_GRSK
 2021-10-28T06:00:00, fakarich, 45, WLMR_GRSK
 2021-10-28T06:00:00, doni, 56, PTR_GRSK
 2021-10-28T06:00:00, doni, 73, SMN_GRSK
 2021-10-28T06:00:00, indra, 82, PTR_GRSK
 2021-10-28T06:00:00, brian, 90, SMN_GRSK
 2021-10-28T06:00:00, brian, 34, WLMR_GRSK
 2021-10-28T06:00:00, indra, 109, SMN_GRSK
 2021-10-28T06:00:00, indra, 99, WLMR_GRSK
 2021-10-28T06:00:00, fakarich, 110, SMN_GRSK
 2021-10-28T06:00:00, fakarich, 88, WLMR_GRSK
 2021-11-10T06:00:00, brian, 66, SMN_GRSK
 2021-11-10T06:00:00, fakarich, 98, PTR_GRSK
```

- **Accessing Auxiliary Input Data**

1. Membuat Xquery dengan nama “accessing.xq”



2. Menjalankan Xquery diatas dengan syntax

```
[oracle@bigdatalite mytools]$ hadoop jar $OXH_HOME/lib/oxh.jar accessing.xq -output ./mydata/myoutaccessing -print
```

Hasilnya:

```
23/03/25 13:49:55 INFO hadoop.xquery: Finished exe
datalite.localdomain:8020/user/oracle/mydata/myout
brian 7
doni 12
fakarich 8
indra 13
```

- **Calling a Custom Java Function from Xquery**
  1. Membuat Xquery dengan nama “javafunction.xq”

```
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))
```

  2. Menjalankan Xquery diatas dengan syntax

```
[oracle@bigdatalite mytools]$ hadoop jar $OXH_HOME/lib/oxh.jar javafunction.xq -output ./mydata/myoutjavafunction -print
```

Hasilnya:

```
23/03/25 21:01:15 INFO hadoop.xquery: Finished executing "javafunction.xq". Output path: "hdfs://bigdatalite.localdomain:8020/us
indra,26,Indra Kenz
fakarich,31,Fakar Suhartami
doni,24,Doni Salmanan
brian,27,Brian Edgar Nababan
[oracle@bigdatalite mytools]$
```

- **Using User-defined XQuery Library Modules and XML Schemas**
  1. Membuat Xquery dengan nama “mytools1.xq” dam “mytools2.xq”

```
File  Edit  View  Search  Terminal  Help
 GNU nano 2.0.9                                  File: mytools1.xq

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

```
File  Edit  View  Search  Terminal  Help
 GNU nano 2.0.9                                  File: mytools2.xq

import module namespace mytools = "urn:mytools" at "mytools.xq";
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))
```

  2. Membuat direktori

```
[oracle@bigdatalite ~]$ mkdir mytool
[oracle@bigdatalite ~]$ mv mytool1.xq mytool2.xq mytool
[oracle@bigdatalite ~]$ cd mytool
[oracle@bigdatalite mytool]$ ls
mytool1.xq  mytool2.xq
[oracle@bigdatalite mytool]$
```

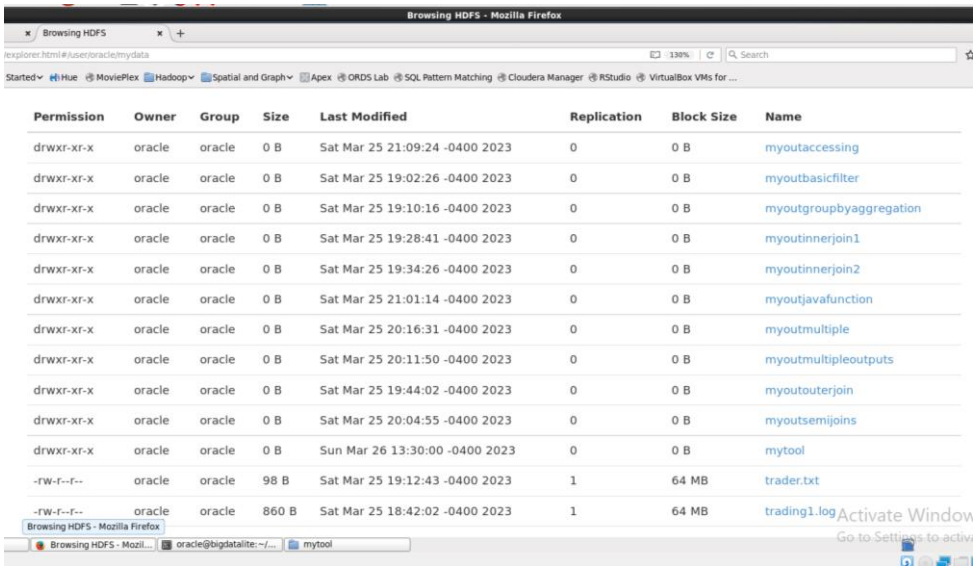
  3. Menjalankan Xquery diatas dengan syntax

```
[oracle@bigdatalite mytool]$ hadoop jar $OXH_HOME/lib/oxh.jar -files mytool1.xq mytoolI2.xq -output ./mydata/mytool -
print
```

Hasilnya :

```
                Bytes Written=0
23/03/26 13:30:02 INFO hadoop.xquery: Finished executing "mytoolI2.xq". Output path: "hdfs://bigdatalite.localdomain:
8020/user/oracle/mydata/mytool"
indra,26,Indra Kenz
fakarich,31,Fakar Suhartami
doni,24,Doni Salmanan
brian,27,Brian Edgar Nababan
[oracle@bigdatalite mytool]$
```

Output keseluruhan pada HDFS :



Genap

- **Xquery Sederhana**
  1. Mengubah file bahanpokok1.log dan bahanpokok2.log kedalam file .txt
  2. Membuka file dengan perintah nano kemudian menyimpannya

3. Simpan file kedalam hdfs

```
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal bahanpokok1.txt
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal bahanpokok2.txt
```

4. Membuat direktori untuk menyimpan output

```
[oracle@bigdatalite ~]$ hadoop fs -mkdir -p /user/mydata2
[oracle@bigdatalite ~]$
```

5. Membuat file gabungan

6. Menjalankan Xquery dengan syntax

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar gabungan_bahan.xq -output ./mydata2/myout1a -print
```

Hasilnya:

```
23/03/26 11:10:25 INFO hadoop.xquery: Finished executing "gabungan_bahan.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata2/myout1a"
 2021-11-28T06:00:00, jateng, 50600, minyak_goreng,in class
 2021-11-28T06:00:00, pertalite, 7632, pertalite,in class
 2021-11-28T06:00:00, jabar, 7642, pertalite,in class
 2021-11-28T06:00:00, sumbar, 2455, telur,in class
 2021-11-28T06:00:00, sumut, 50400, minyak_goreng,in class
 2021-11-28T06:00:00, sumbar, 7630, pertalite,in class
 2021-11-28T06:00:00, jatim, 24040, telur,in class
 2021-11-28T06:00:00, jatim, 50300, minyak_goreng,in class
 2021-11-28T06:00:00, jatim, 24050, telur,in class
 2021-11-28T06:00:00, jateng, 50300, minyak_goreng,in class
 2021-11-28T06:00:00, jabar, 23430, telur,in class
 2021-11-28T06:00:00, jateng, 50100, minyak_goreng,in class
 2021-11-28T06:00:00, jatim, 7600, pertalite,in class
 2021-11-28T06:00:00, sumut, 12510, pertamax,in class
 2021-11-28T06:00:00, jatim, 24010, telur,in class
 2021-11-28T06:00:00, sumbar, 50300, minyak_goreng,in class
 2021-11-28T06:00:00, sumut, 7630, pertalite,in class
 2021-11-28T06:00:00, sumbar, 12500, pertamax,in class
 2021-11-28T06:00:00, sumut, 24500, telur,in class
 2021-11-28T06:00:00, sumut, 50100, minyak_goreng,in class
 2021-09-28T06:00:00, jatim, 24000, telur,in class
 2021-09-28T06:00:00, jateng, 50600, minyak_goreng,in class
 2021-09-28T06:00:00, jabar, 7642, pertalite,in class
 2021-09-28T06:00:00, sumbar, 12560, pertamax,in class
 2021-09-28T06:00:00, sumut, 7633, pertalite,in class
 2021-09-28T06:00:00, sumbar, 12450, pertamax,in class
 2021-09-28T06:00:00, jateng, 24060, telur,in class
 2021-09-28T06:00:00, jatim, 50300, minyak_goreng,in class
 2021-09-28T06:00:00, jatim, 7649, pertalite,in class
 2021-09-28T06:00:00, jateng, 12460, pertamax,in class
 2021-09-28T06:00:00, jabar, 23450, telur,in class
 2021-09-28T06:00:00, jatim, 50200, minyak_goreng,in class
 2021-09-28T06:00:00, jabar, 7620, pertalite,in class
 2021-09-28T06:00:00, jabar, 12520, pertamax,in class
 2021-09-28T06:00:00, jabar, 24020, telur,in class
 2021-09-28T06:00:00, sumbar, 50300, minyak_goreng,in class
 2021-09-28T06:00:00, sumut, 7630, pertalite,in class
 2021-09-28T06:00:00, jatim, 12600, pertamax,in class
 2021-09-28T06:00:00, sumut, 24500, telur,in class
 2021-09-28T06:00:00, sumbar, 50200, minyak_goreng,in class
[ora
```


- **XQuery Basic Filtering**
  1. Mengcopy file bahanpokok1.log dan bahanpokok2.log ekdalam direktori hadoop

```
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal bahanpokok1.log /user/oracle/mydata2
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal bahanpokok2.log /user/oracle/mydata2
[oracle@bigdatalite ~]$
```


  2. Membuat file bernama “basicfilter2.xq”

```
 GNU nano 2.0.9                    File: basicfilterr.xq                              Modified

import module "oxh:text";
for $line in
text:collection("mydata2/bahanpokok*.log")
let $split := fn:tokenize($line, "\s*,\s*")
where $split[2] eq "jabar"
return text:put($line)
```

3. **Menjalankan Xquery dengan syntax**

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar basicfilterr.xq -output ./mydata2/myoutbasicfilterr -print
```

Hasilnya :

```
23/03/26 11:34:48 INFO hadoop.xquery: Finished executing "basicfilterr.xq". Output path: "hdfs://bigdatalite.localdom
ain:8020/user/oracle/mydata2/myoutbasicfilterr"
 2021-11-28T06:00:00, jabar, 7642, pertalite
 2021-11-28T06:00:00, jabar, 23430, telur
 2021-09-28T06:00:00, jabar, 7642, pertalite
 2021-09-28T06:00:00, jabar, 23450, telur
 2021-09-28T06:00:00, jabar, 7620, pertalite
 2021-09-28T06:00:00, jabar, 12520, pertamax
 2021-09-28T06:00:00, jabar, 24020, telur
[oracle@bigdatalite ~]$
```


- Group by and Aggregation
  1. Membuat file bernama “groupby.xq”

```
 GNU nano 2.0.9                    File: groupby.xq                               Modified

import module "oxh:text";
for $line in text:collection("mydata2/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))
```

  2. Menjalankan Xquery dengan syntax

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar groupby.xq -output ./mydata2/myoutgroupby -print
```

Hasilnya:

```
23/03/26 11:42:39 INFO hadoop.xquery: Finished executing "groupby.xq". Output path: "hdfs://bigdatalite.localdomain:8
020/user/oracle/mydata2/myoutgroupby"
2021-09-28 => 20
2021-11-28 => 20
[oracle@bigdatalite ~]$
```


- **Inner Joins**
  1. Mengcopy file “provinsi.txt” kedalam HDFS

```
[oracle@bigdatalite ~]$ hdfs dfs -copyFromLocal provinsi.txt /user/oracle/mydata2
```

  2. Membuat file bernama “inerjoin1.xq” dan “inerjoin2.xq”
     - Inerjoin1.xq

```
 GNU nano 2.0.9                    File: inerjoinn1.xq                            Modified

import module "oxh:text";
for $userLine in text:collection("mydata2/provinsi.txt")
let $userSplit := fn:tokenize($userLine, "\s*:\s*")
let $userId := $userSplit[1]
let $userAge := xs:integer($userSplit[3][. castable as xs:integer])

for $visitLine in text:collection("mydata2/bahanpokok*.log")
let $visitSplit := fn:tokenize($visitLine, "\s*,\s*")
let $visitUserId := $visitSplit[2]
where $userId eq $visitUserId and $userAge gt 30
group by $page := $visitSplit[3]
return text:put($page || " " || fn:count($userLine))
```

     - Inejoin2.xq

```
 GNU nano 2.0.9                    File: inerjoin2.xq                              Modified

import module "oxh:text";
for $userLine in text:collection("mydata2/provinsi.txt")
let $userSplit := fn:tokenize($userLine, "\s*:\s*")
let $userId := $userSplit[1]

for $visitLine in text:collection("mydata2/bahanpokok*.log")
[$userId eq fn:tokenize(., "\s*,\s*")[2]]

group by $userId
return text:put($userId || " " || fn:count($visitLine))
```

3. Menjalankan Xquery dengan syntax
   - Inerjoin1.xq

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar inerjoin1.xq -output ./mydata2/myoutinerjoin1 -print
```

   - Inerjoin2.xq

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar inerjoin2.xq -output ./mydata2/myoutinerjoin2 -print
```

Hasilnya:

   - Inerjoin1.xq

```
jabar-pertalite 3
jabar-pertamax 1
jabar-telur 3
jateng-minyak_goreng 4
jateng-pertamax 1
jateng-telur 1
jatim-minyak_goreng 3
jatim-pertalite 2
jatim-pertamax 1
jatim-telur 4
sumbar-minyak_goreng 3
sumbar-pertalite 1
sumbar-pertamax 3
sumbar-telur 1
sumut-minyak_goreng 2
sumut-pertalite 3
sumut-pertamax 1
sumut-telur 2
```

   - Inerjoin2.xq

```
23/03/26 11:56:34 INFO hadoop.xquery: Finished executing "inerjoin2.xq". Output path: "hdfs://bigdatalite.localdomain
:8020/user/oracle/mydata2/myoutinerjoin2"
jabar 7
jateng 6
jatim 10
sumbar 8
sumut 8
[oracle@bigdatalite ~]$
```


- **Left Outer Joins**
  1. Membuat file bernama “outerjoinn.xq”

```
 GNU nano 2.0.9                    File: outerjoinn.xq                             Modified

import module "oxh:text";

for $userLine in text:collection("mydata2/provinsi.txt")
let $userSplit := fn:tokenize($userLine, "\s*:\s*")
let $userId := $userSplit[1]

for $visitLine allowing empty in
text:collection("mydata2/bahanpokok*.log")
[$userId eq fn:tokenize(., "\s*,\s*")[2]]

group by $userId
return text:put($userId || " " || fn:count($visitLine))
```

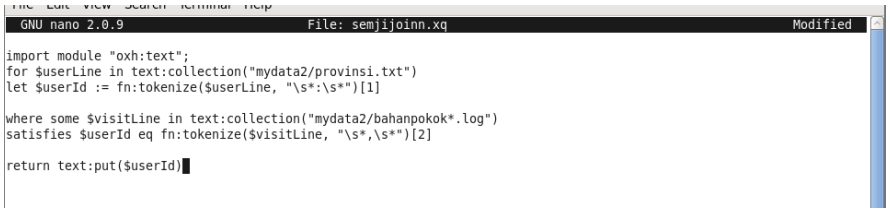
  2. Menjalankan Xquery dengan syntax

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar outerjoinn.xq -output ./mydata2/myoutouterjoinn -print
```

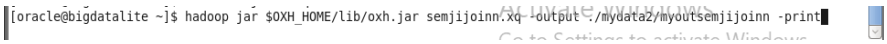
Hasilnya:

```
23/03/26 12:20:00 INFO hadoop.xquery: Finished executing "outerjoinn.xq". Output path: "hdfs://bigdatalite.localdomai
n:8020/user/oracle/mydata2/myoutouterjoinn"
jabar 7
jateng 6
jatim 10
sumbar 8
sumut 8
[oracle@bigdatalite ~]$
```

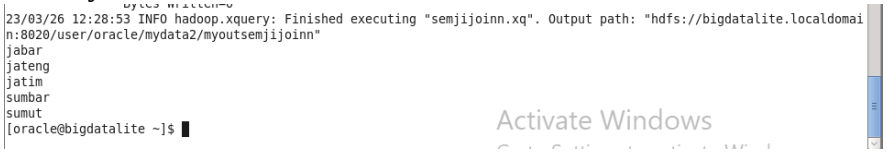
- **Semijoins**
  1. Membuat file bernama “semjijoinn.xq”



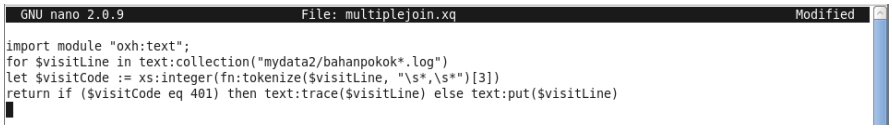
  2. Menjalankan Xquery dengan syntax



Hasilnya:

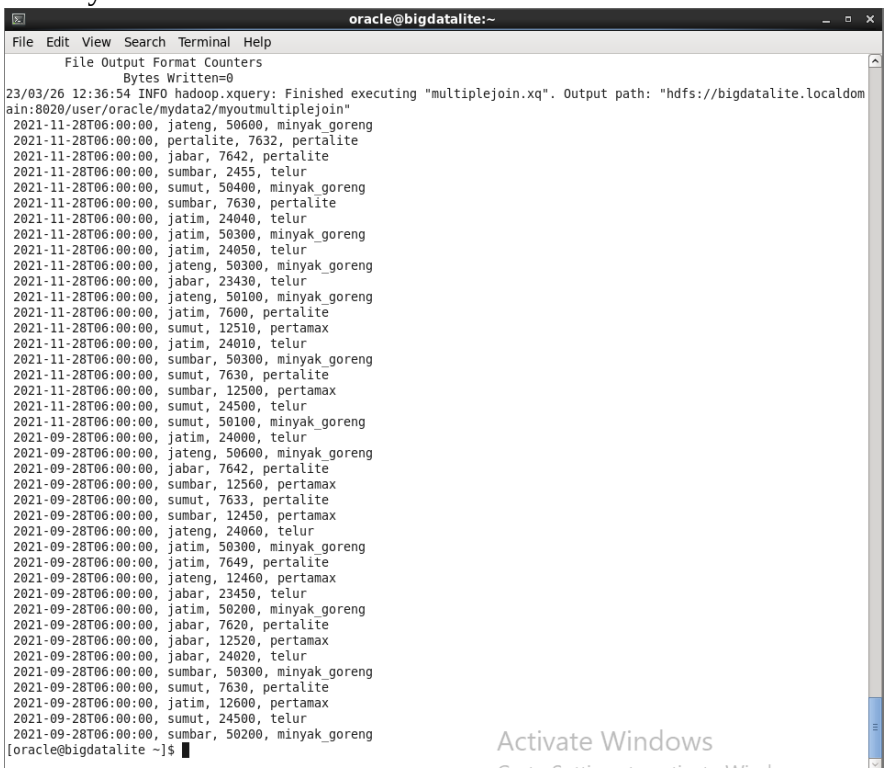


- **Multiple Outputs**
  1. Membuat file bernama “multi.xq”



  2. Menjalankan Xquery dengan syntax

Hasilnya:



- **Accessing Auxiliary Input Data**
  1. Membuat file bernama “accessing.xq”

```
 GNU nano 2.0.9                    File: accessing.xq                              Modified

import module "oxh:text";
for $visitLine in text:collection("mydata2/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))
```

  2. Menjalankan Xquery dengan syntax

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar -files provinsi.txt accessing.xq -output ./mydata2/myoutacce
ssin
```

oracle@bigdatalite:~

Hasilnya :

```
23/03/26 12:44:11 INFO hadoop.xquery: Finished executing "accessing.xq". Output path: "hdfs://bigdatalite.localdomain
:8020/user/oracle/mydata2/myoutaccessing"
jabar 7
jateng 6
jatim 10
sumbar 8
sumut 8
[oracle@bigdatalite ~]$
```

- **Calling a Custom Java Function from Xquery**
  1. Membuat file bernama “javafunc.xq”

```
 GNU nano 2.0.9                    File: javafunc.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*) as
xs:string external;

for $line in text:collection("mydata2/provinsi*.txt")
let $split := fn:tokenize($line, "\s*:\s*")
return text:put(local:string-format("%s,%s,%s", $split))
```

  2. Menjalankan Xquery dengan syntax

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar javafunc.xq -output ./mydata2/myoutjavafunc -print
```

Hasilnya:

```
23/03/26 12:47:50 INFO hadoop.xquery: Finished executing "javafunc.xq". Output path: "hdfs://bigdatalite.localdomain:
8020/user/oracle/mydata2/myoutjavafunc"
sumut,,Sumatera Utara
sumbar,,Sumatera Barat
jabar,,Jawa Barat
jatim,,Jawa Timur
jateng,,Jawa Tengah
[oracle@bigdatalite ~]$
```

- **Using User-defined XQuery Library Modules and XML Schemas**
  1. Membuat file bernama “tools1.xq” dan “tools2.xq”
     - Tools1.xq

```
 GNU nano 2.0.9                    File: tools1.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:anyAtomicType*) as
xs:string external;
```

     - Tools2.xq

```
 GNU nano 2.0.9                    File: tools2.xq

import module namespace mytools = "urn:mytools" at "tools1.xq";
import module "oxh:text";

for $line in text:collection("mydata2/provinsi*.txt")
let $split := fn:tokenize($line, "\s*:\s*")
return text:put(mytools:string-format("%s,%s,%s", $split))
```

2. Membuat folder baru dengan nama mytools2 dan memindahkan file tools1.xq dan tools2.xq kedalam direktori tersebut
3. Menjalankan Xquery dengan syntax

```
[oracle@bigdatalite mytools2]$ hadoop jar $OXH_HOME/lib/oxh.jar -files tools1.xq tools2.xq -output ./mydata2/myoutmyt
ools2 -print
```

Hasilnya:

```
23/03/26 13:08:34 INFO hadoop.xquery: Finished executing "tools2.xq". Output path: "hdfs://bigdatalite.localdomain:80
20/user/oracle/mydata2/myoutmytools2"
sumut,,Sumatera Utara
sumbar,,Sumatera Barat
jabar,,Jawa Barat
jatim,,Jawa Timur
jateng,,Jawa Tengah
[oracle@bigdatalite mytools2]$
```


Secara keseluruhan hasil diatas dapat dilihat pada direktori hdfs