

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:

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

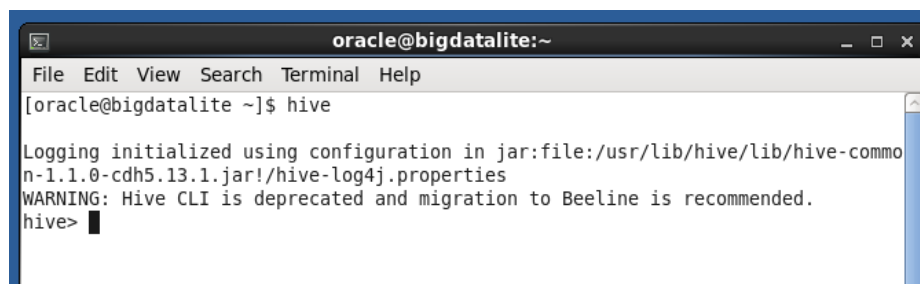
di <https://drive.google.com/drive/folders/182b5TikHcqCe2vAzgfabaNNjAfG5Qh6s?usp=sharing>. Lalu analisislah menggunakan bentuk-bentuk Xquery transformation yang sesuai

JAWABAN

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

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

Sebelum menghitung nilai-nilainya kita harus masuk dulu ke dalam hive seperti gambar berikut :



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ hive  
  
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-common-1.1.0-cdh5.13.1.jar!/hive-log4j.properties  
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.  
hive>
```

- **Suhu Maksimum**

Untuk menghitung nilai suhu maksimum nya menggunakan syntax berikut :

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

Output :

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
OK  
1901 999  
1902 328  
1903 999  
1904 294  
1905 328  
1906 294  
1907 999  
1908 378  
1909 999  
1910 999  
1911 999  
1912 411  
1913 999  
1914 999  
1915 999  
1916 289  
1917 478  
1918 999  
1919 999  
1920 344  
1921 999  
1922 999  
1923 394  
1923 394  
1924 456  
1925 378  
1926 999  
1927 999  
1928 999  
1929 999  
1930 999  
1931 999  
1932 999  
Time taken: 24.929 seconds, Fetched: 32 row(s)  
hive>
```

- **Suhu Minimum**

Untuk menghitung nilai suhu minimum nya menggunakan syntax berikut :

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

Output:

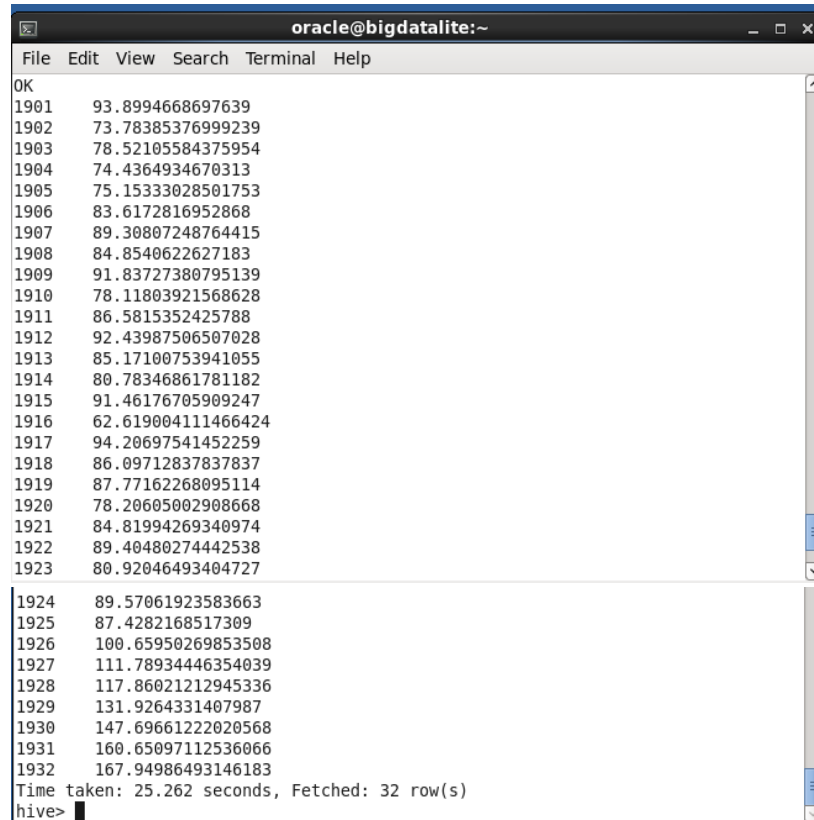
```
File Edit View Search Terminal Help  
1901 0  
1902 0  
1903 0  
1904 0  
1905 0  
1906 0  
1907 0  
1908 0  
1909 0  
1910 0  
1911 0  
1912 0  
1913 0  
1914 0  
1915 0  
1916 0  
1917 0  
1918 0  
1919 0  
1920 0  
1921 0  
1922 0  
1923 0  
1924 0  
1924 0  
1925 0  
1926 0  
1927 0  
1928 0  
1929 0  
1930 0  
1931 0  
1932 0  
Time taken: 24.403 seconds, Fetched: 32 row(s)  
hive>
```

- **Rata-Rata**

Untuk menghitung nilai rata-rata nya menggunakan syntax berikut :

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

Output:



```
oracle@bigdatalite:~
File Edit View Search Terminal Help
OK
1901 93.8994668697639
1902 73.78385376999239
1903 78.52105584375954
1904 74.4364934670313
1905 75.15333028501753
1906 83.6172816952868
1907 89.30807248764415
1908 84.8540622627183
1909 91.83727380795139
1910 78.11803921568628
1911 86.5815352425788
1912 92.43987506507028
1913 85.17100753941055
1914 80.78346861781182
1915 91.46176705909247
1916 62.619004111466424
1917 94.20697541452259
1918 86.09712837837837
1919 87.77162268095114
1920 78.20605002908668
1921 84.81994269340974
1922 89.40480274442538
1923 80.92046493404727
1924 89.57061923583663
1925 87.4282168517309
1926 100.65950269853508
1927 111.78934446354039
1928 117.86021212945336
1929 131.9264331407987
1930 147.69661222020568
1931 160.65097112536066
1932 167.94986493146183
Time taken: 25.262 seconds, Fetched: 32 row(s)
hive>
```

- **Varian**

Untuk menghitung nilai varian nya menggunakan syntax berikut :

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

Output:

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
Total MapReduce CPU Time Spent: 3 seconds 510 msec  
OK  
1901 4763.648994384471  
1902 2952.782069839996  
1903 9037.802806575155  
1904 2813.311904325309  
1905 3649.533264705588  
1906 4165.583705727371  
1907 4402.526544756256  
1908 4209.395103088001  
1909 7828.16583295886  
1910 4405.390903344865  
1911 6511.908057117325  
1912 5485.1690815044385  
1913 5929.42917542993  
1914 4895.375316058272  
1915 4994.930673343384  
1916 2821.9215392548913  
1917 5113.103187478508  
1918 6278.702559321321  
1919 4649.827645211168  
1920 3998.41932349021  
1921 4025.389585096988  
1922 9609.395425761575  
1923 3350.742139629734  
1924 4091.520111737748  
1925 4591.046414782489  
1926 13678.42964755529  
1927 24760.95881599549  
1928 32968.09223525558  
1929 28532.437138069305  
1930 35426.33148434418  
1931 44478.2234253197  
1932 59660.99342985473  
Time taken: 23.019 seconds, Fetched: 32 row(s)  
hive>
```

- **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:

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
Total MapReduce CPU Time Spent: 3 seconds 570 msec  
OK  
1901 69.01919294214089  
1902 54.339507449368696  
1903 95.06735931209595  
1904 53.040662744024125  
1905 60.41136701569985  
1906 64.54133331228424  
1907 66.35153762164262  
1908 64.87985128749912  
1909 88.47692260108768  
1910 66.37311883093084  
1911 80.69639432538064  
1912 74.06192734127595  
1913 77.00278680301078  
1914 69.96695874524112  
1915 70.67482347585585  
1916 53.12176144721569  
1917 71.50596609709226  
1918 79.23826448958432  
1919 68.1896447065914  
1920 63.233055623544004  
1921 63.445957988645645  
1922 98.02752381735232
```

```

1922  90.02732301733232
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:

```
Total MapReduce CPU Time Spent: 6 seconds 580 msec
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:

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
Total MapReduce CPU Time Spent: 2 seconds 630 msec  
OK  
1901 6565  
1902 6565  
1903 6554  
1904 6582  
1905 6561  
1906 5474  
1907 5463  
1908 6585  
1909 7571  
1910 7650  
1911 7647  
1912 7684  
1913 8754  
1914 8747  
1915 8749  
1916 2189  
1917 8745  
1918 8288  
1919 7654  
1920 8595  
1921 8725  
1922 8745  
1922 8745  
1923 7657  
1924 7590  
1925 7655  
1926 10376  
1927 4073  
1928 7354  
1929 32202  
1930 89262  
1931 258566  
1932 219148  
Time taken: 20.703 seconds, Fetched: 32 row(s)  
hive>
```

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:

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
Total MapReduce CPU Time Spent: 3 seconds 430 msec  
OK  
1901 NULL  
1902 NULL  
1903 NULL  
1904 NULL  
1905 NULL  
1906 NULL  
1907 NULL  
1908 NULL  
1909 NULL  
1910 NULL  
1911 NULL  
1912 NULL  
1913 NULL  
1914 NULL  
1915 NULL  
1916 NULL  
1917 NULL  
1918 NULL  
1919 NULL  
1920 NULL  
1921 NULL  
1922 NULL
```

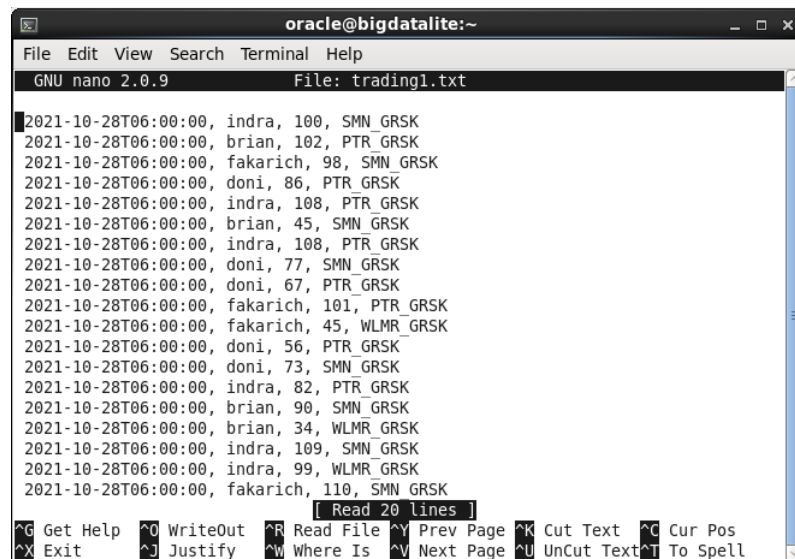
Nilai null pada data tidak akan mempengaruhi hasil korelasi antara suhu maksimum dan suhu minimum pada setiap tahun. Jadi tidak ada korelasi antara suhu maksimum dan suhu minimum

2. Analisis menggunakan bentuk-bentuk Xquery transformation

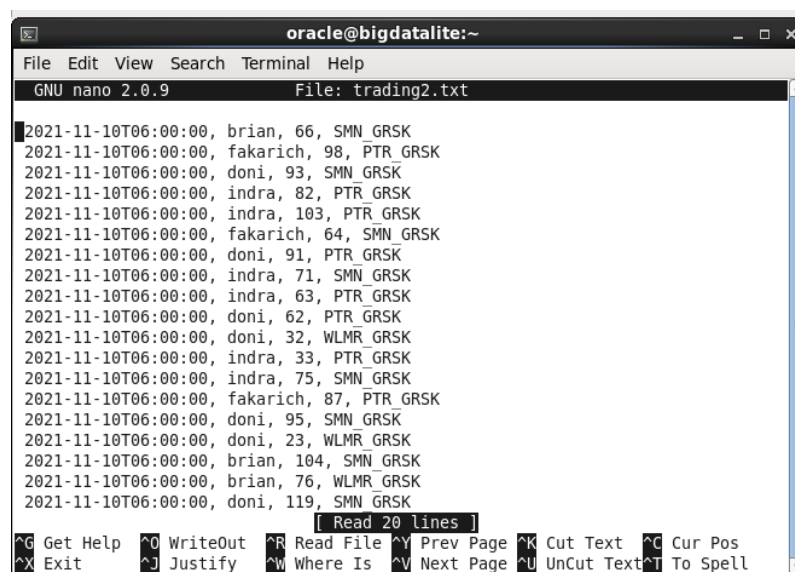
Ganji

▪ Xquery Sederhana

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



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: trading1.txt  
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  
[ Read 20 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```



```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
GNU nano 2.0.9 File: trading2.txt  
2021-11-10T06:00:00, brian, 66, SMN_GRSK  
2021-11-10T06:00:00, fakarich, 98, PTR_GRSK  
2021-11-10T06:00:00, doni, 93, SMN_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, SMN_GRSK  
2021-11-10T06:00:00, doni, 91, PTR_GRSK  
2021-11-10T06:00:00, indra, 71, SMN_GRSK  
2021-11-10T06:00:00, indra, 63, PTR_GRSK  
2021-11-10T06:00:00, doni, 62, PTR_GRSK  
2021-11-10T06:00:00, doni, 32, WLMR_GRSK  
2021-11-10T06:00:00, indra, 33, PTR_GRSK  
2021-11-10T06:00:00, indra, 75, SMN_GRSK  
2021-11-10T06:00:00, fakarich, 87, PTR_GRSK  
2021-11-10T06:00:00, doni, 95, SMN_GRSK  
2021-11-10T06:00:00, doni, 23, WLMR_GRSK  
2021-11-10T06:00:00, brian, 104, SMN_GRSK  
2021-11-10T06:00:00, brian, 76, WLMR_GRSK  
2021-11-10T06:00:00, doni, 119, SMN_GRSK  
[ Read 20 lines ]  
^G Get Help ^O WriteOut ^R Read File ^Y Prev Page ^K Cut Text ^C Cur Pos  
^X Exit ^J Justify ^W Where Is ^V Next Page ^U UnCut Text ^T To Spell
```

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



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

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

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

Hasilnya :

```
File Edit View Search Terminal Help
File Input Format Counters
Bytes Read=1709
File Output Format Counters
Bytes Written=0
23/03/25 10:30:40 INFO hadoop.xquery: Finished executing "trading.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/myout"
2021-10-28T06:00:00, Indra, 100, SMN GRSkin class
2021-10-28T06:00:00, Brian, 102, PTR GRSkin class
2021-10-28T06:00:00, Fakarich, 90, SMN GRSkin class
2021-10-28T06:00:00, doni, 86, PTR GRSkin class
2021-10-28T06:00:00, Indra, 100, PTR GRSkin class
2021-10-28T06:00:00, Brian, 45, SMN GRSkin class
2021-10-28T06:00:00, Indra, 100, PTR GRSkin class
2021-10-28T06:00:00, doni, 77, SMN GRSkin class
2021-10-28T06:00:00, doni, 87, PTR GRSkin class
2021-10-28T06:00:00, Fakarich, 101, PTR GRSkin class
2021-10-28T06:00:00, Fakarich, 45, WLMR GRSkin class
2021-10-28T06:00:00, doni, 56, PTR GRSkin class
2021-10-28T06:00:00, doni, 73, SMN GRSkin class
2021-10-28T06:00:00, Indra, 82, PTR GRSkin class
2021-10-28T06:00:00, Brian, 90, SMN GRSkin class
2021-10-28T06:00:00, Brian, 34, WLMR GRSkin class
2021-10-28T06:00:00, Indra, 100, SMN GRSkin class
2021-10-28T06:00:00, Indra, 99, WLMR GRSkin class
2021-10-28T06:00:00, Fakarich, 110, SMN GRSkin class
2021-10-28T06:00:00, Fakarich, 88, WLMR GRSkin class
2021-11-10T06:00:00, Brian, 66, SMN GRSkin class
2021-11-10T06:00:00, Fakarich, 90, PTR GRSkin class
2021-11-10T06:00:00, doni, 93, SMN GRSkin class
2021-11-10T06:00:00, Indra, 103, PTR GRSkin class
2021-11-10T06:00:00, Fakarich, 64, SMN GRSkin class
2021-11-10T06:00:00, doni, 91, PTR GRSkin class
2021-11-10T06:00:00, Indra, 71, SMN GRSkin class
2021-11-10T06:00:00, Indra, 63, PTR GRSkin class
2021-11-10T06:00:00, doni, 62, PTR GRSkin class
2021-11-10T06:00:00, doni, 32, WLMR GRSkin class
2021-11-10T06:00:00, Indra, 33, PTR GRSkin class
2021-11-10T06:00:00, Indra, 75, SMN GRSkin class
2021-11-10T06:00:00, Fakarich, 87, PTR GRSkin class
2021-11-10T06:00:00, doni, 95, SMN GRSkin class
2021-11-10T06:00:00, doni, 23, WLMR GRSkin class
2021-11-10T06:00:00, Brian, 104, SMN GRSkin class
2021-11-10T06:00:00, Brian, 76, WLMR GRSkin class
2021-11-10T06:00:00, doni, 119, SMN GRSkin class
2021-11-10T06:00:00, Indra, 85, WLMR GRSkin class
```

■ 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’

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

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

4. Menjalankan Xquery diatas dengan syntax

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

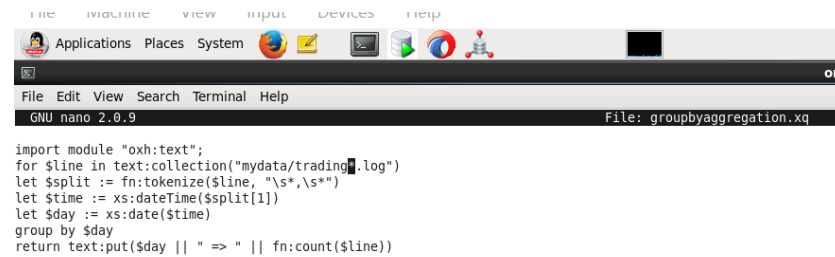
Hasilnya:

```
1/03/25 19:02:20 INFO hadoop.xquery: Finished executing "basicfilter.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata/myoutbasicfilter"
1021-10-28T06:00:00, brian, 102, PTR GRSK
1021-10-28T06:00:00, brian, 45, SMN GRSK
1021-10-28T06:00:00, brian, 90, SMN GRSK
1021-10-28T06:00:00, brian, 34, WLMR GRSK
1021-11-10T06:00:00, brian, 66, SMN GRSK
1021-11-10T06:00:00, brian, 104, SMN GRSK
1021-11-10T06:00:00, brian, 76, WLMR GRSK
[oracle@bigdatalite ~]$
```

■ 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



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

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))
```

2. Menjalankan Xquery diatas dengan syntax

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

Hasilnya:

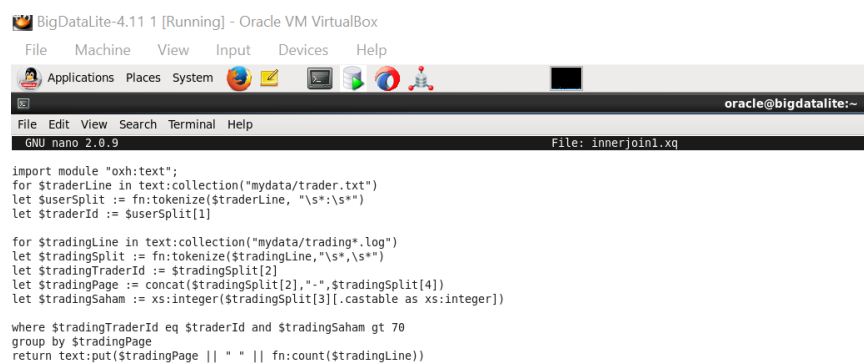
```
23/03/25 19:10:18 INFO hadoop.xquery: Finished executing "grou
2021-10-28 => 20
2021-11-10 => 20
[oracle@bigdatalite ~]$
```

■ 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

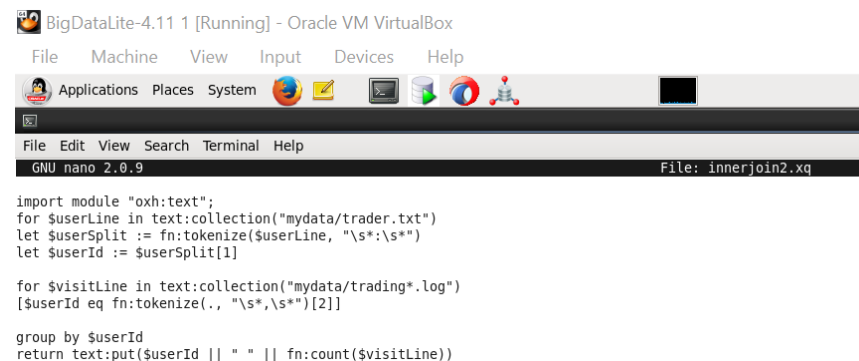


```
BigDataLite-4.11 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
oracle@bigdatalite:~
File Edit View Search Terminal Help
GNU nano 2.0.9 File: innerjoin1.xq

import module "oxh:text";
for $traderLine in text:collection("mydata/trader.txt")
let $userSplit := fn:tokenize($traderLine, "\s*\s*")
let $traderId := $userSplit[1]

for $tradingLine in text:collection("mydata/trading*.log")
let $tradingSplit := fn:tokenize($tradingLine, "\s*\s*")
let $tradingTraderId := $tradingSplit[2]
let $tradingPage := concat($tradingSplit[2], "-", $tradingSplit[4])
let $tradingSaham := xs:integer($tradingSplit[3][.castable as xs:integer])

where $tradingTraderId eq $traderId and $tradingSaham gt 70
group by $tradingPage
return text:put($tradingPage || " " || fn:count($tradingLine))
```



```
BigDataLite-4.11 1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places System
oracle@bigdatalite:~
File Edit View Search Terminal Help
GNU nano 2.0.9 File: innerjoin2.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 $visitLine in text:collection("mydata/trading*.log")
[$userId eq fn:tokenize(., "\s*\s*")[2]]

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

3. Menjalankan Xquery diatas dengan syntax

Innerjoin1.xq

```
oracle@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". Out
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:~

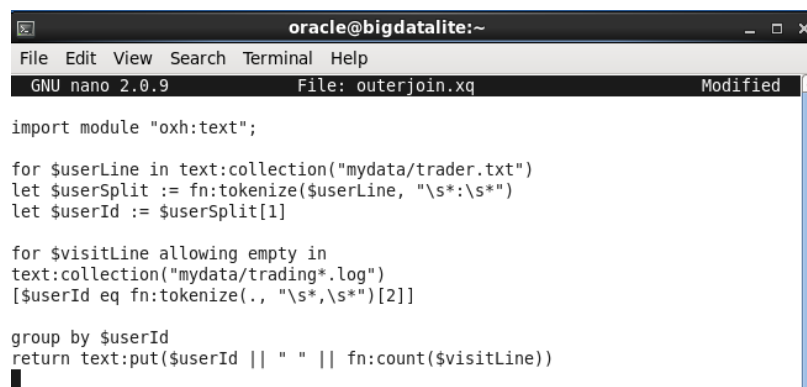
Hasilnya innerjoin2.xq:

```
23/03/25 19:34:28 INFO hadoop.xquery: Finished executing "innerjoin2.xq". Out
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’.



```
oracle@bigdatalite:~
File Edit View Search Terminal Help
GNU nano 2.0.9 File: outerjoin.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]


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))
```

2. Menjalankan Xquery diatas dengan syntax

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

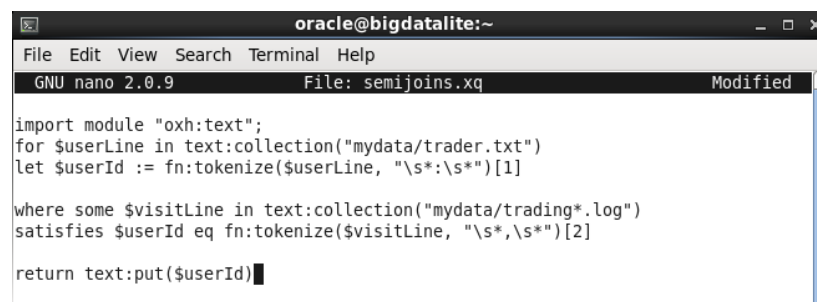
XQuery tersebut berguna untuk menampilkan user siapa saja yang pernah mengakses

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.



```

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

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)

```

2. Menjalankan Xquery diatas dengan syntax

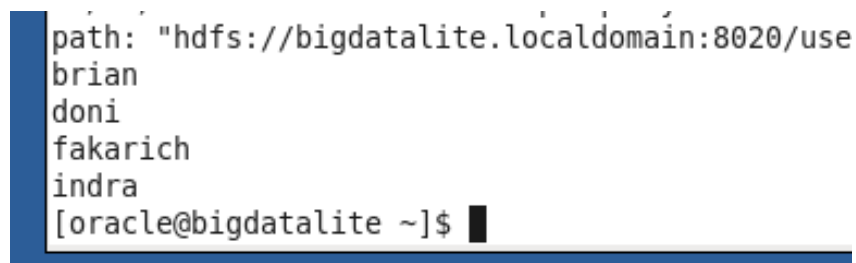


```

[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar semijoins.xq -output ./mydata/myoutsemijoins -print

```

Hasilnya :



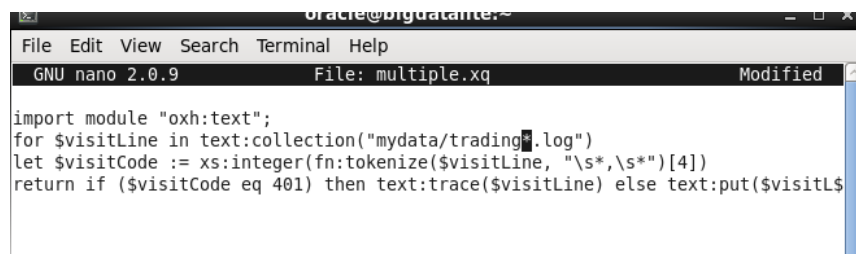
```

path: "hdfs://bigdatalite.localdomain:8020/use
brian
doni
fakarich
indra
[oracle@bigdatalite ~]$

```

- **Multiple Outputs**

1. Membuat Xquery dengan nama “multipleoutputs.xq”



```

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

import module "oxh:text";
for $visitLine in text:collection("mydata/trading*.log")
let $visitCode := xs:integer(fn:tokenize($visitLine, "\s*:\s*")[4])
return if ($visitCode eq 401) then text:trace($visitLine) else text:put($visitL$

```

2. Menjalankan Xquery diatas dengan syntax

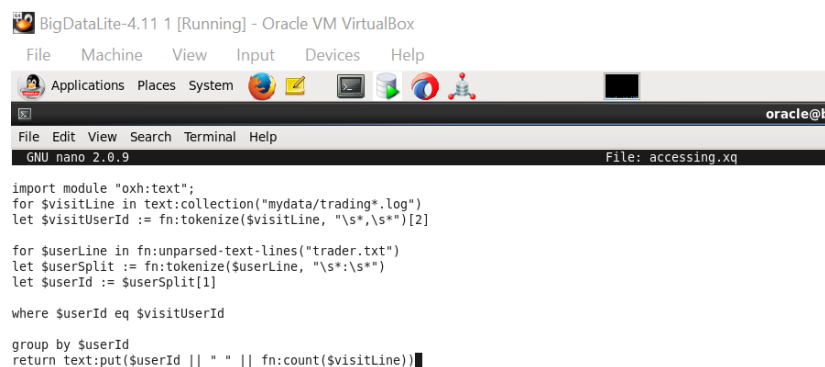
```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar multiple.xq -output ./mydata/myoutmultiple -print
```

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”



```
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))
```

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
23/03/25 21:00:50 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1.edb5.0.0.mr3.0.mr3) Copyright (c) 2023
```

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 "mytools1.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

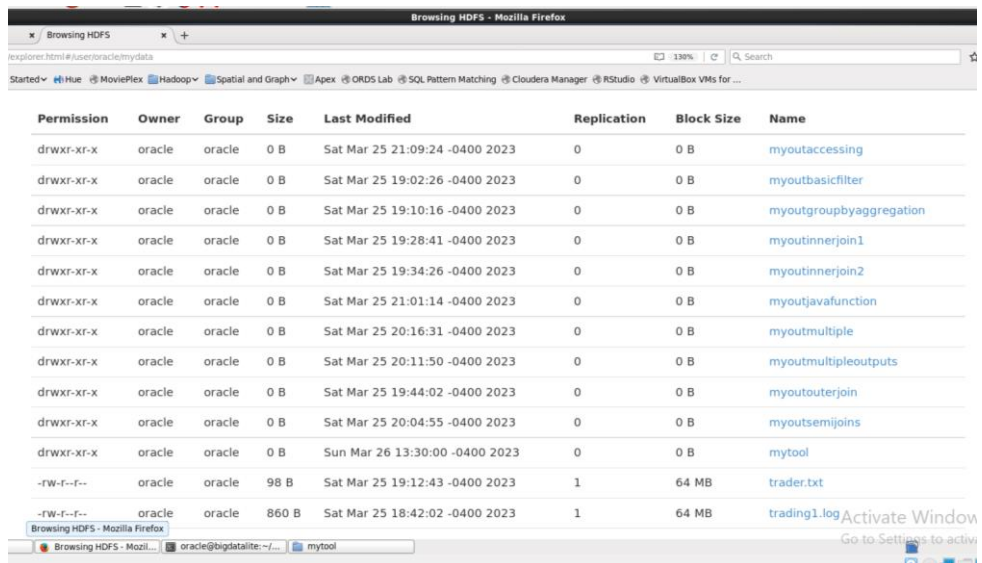
```
mytool1.xq mytool2.xq
[oracle@bigdatalite mytool]$ hadoop jar $OXH_HOME/lib/oxh.jar -files mytool1.xq mytool2.xq -output ./mydata/mytool -
print
23/03/26 13:30:47 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.0.1 (build 4.0.1.edb5.0.0.mr3.0.mr3) Copyright (c) 2023
```

Hasilnya :

```
Bytes Written=0
23/03/26 13:30:02 INFO hadoop.xquery: Finished executing "mytool2.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]$
```

Activate Windows

Output keseluruhan pada HDFS :



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 21:09:24 -0400 2023	0	0 B	myoutaccessing
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 19:02:26 -0400 2023	0	0 B	myoutbasicfilter
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 19:10:16 -0400 2023	0	0 B	myoutgroupbyaggregation
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 19:28:41 -0400 2023	0	0 B	myoutinnerjoin1
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 19:34:26 -0400 2023	0	0 B	myoutinnerjoin2
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 21:01:14 -0400 2023	0	0 B	myoutjavafunction
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 20:16:31 -0400 2023	0	0 B	myoutmultiple
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 20:11:50 -0400 2023	0	0 B	myoutmultipleoutputs
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 19:44:02 -0400 2023	0	0 B	myoutouterjoin
drwxr-xr-x	oracle	oracle	0 B	Sat Mar 25 20:04:55 -0400 2023	0	0 B	myoutsemijoins
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 13:30:00 -0400 2023	0	0 B	mytool
-rw-r--r--	oracle	oracle	98 B	Sat Mar 25 19:12:43 -0400 2023	1	64 MB	trader.txt
-rw-r--r--	oracle	oracle	860 B	Sat Mar 25 18:42:02 -0400 2023	1	64 MB	trading1.log

Genap

▪ Xquery Sederhana

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

```
GNU nano 2.0.9 File: bahanpokok1.txt
2021-09-28T06:00:00, jatim, 24000, telur
2021-09-28T06:00:00, jateng, 50600, minyak goreng
2021-09-28T06:00:00, jabar, 7642, pertalite
2021-09-28T06:00:00, sumbar, 12560, pertamax
2021-09-28T06:00:00, sumut, 7633, pertalite
2021-09-28T06:00:00, sumbar, 12450, pertamax
2021-09-28T06:00:00, jateng, 24060, telur
2021-09-28T06:00:00, jatim, 50300, minyak goreng
2021-09-28T06:00:00, jatim, 7649, pertalite
2021-09-28T06:00:00, jateng, 12460, pertamax
2021-09-28T06:00:00, jabar, 23450, telur
2021-09-28T06:00:00, jatim, 50200, minyak goreng
2021-09-28T06:00:00, jabar, 7620, pertalite
2021-09-28T06:00:00, jabar, 12520, pertamax
2021-09-28T06:00:00, jabar, 24020, telur
2021-09-28T06:00:00, sumbar, 50300, minyak goreng
2021-09-28T06:00:00, sumut, 7630, pertalite
2021-09-28T06:00:00, jatim, 12600, pertamax
2021-09-28T06:00:00, sumut, 24500, telur
2021-09-28T06:00:00, sumbar, 50200, minyak goreng
```

```
GNU nano 2.0.9 File: bahanpokok2.txt
2021-11-28T06:00:00, jateng, 50600, minyak goreng
2021-11-28T06:00:00, pertalite, 7632, pertalite
2021-11-28T06:00:00, jabar, 7642, pertalite
2021-11-28T06:00:00, sumbar, 2455, telur
2021-11-28T06:00:00, sumut, 50400, minyak goreng
2021-11-28T06:00:00, sumbar, 7630, pertalite
2021-11-28T06:00:00, jatim, 24040, telur
2021-11-28T06:00:00, jatim, 50300, minyak goreng
2021-11-28T06:00:00, jatim, 24050, telur
2021-11-28T06:00:00, jateng, 50300, minyak goreng
2021-11-28T06:00:00, jabar, 23430, telur
2021-11-28T06:00:00, jateng, 50100, 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, 50300, minyak goreng
2021-11-28T06:00:00, sumut, 7630, pertalite
2021-11-28T06:00:00, sumbar, 12500, pertamax
2021-11-28T06:00:00, sumut, 24500, telur
2021-11-28T06:00:00, sumut, 50100, minyak goreng
```

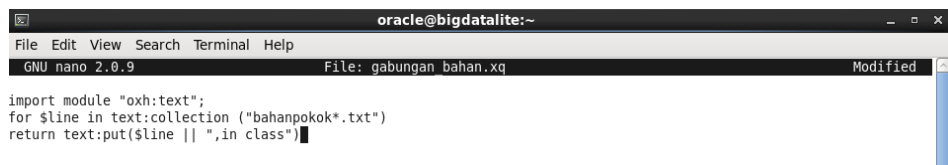

3. Simpan file kedalam hdfs

```
[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

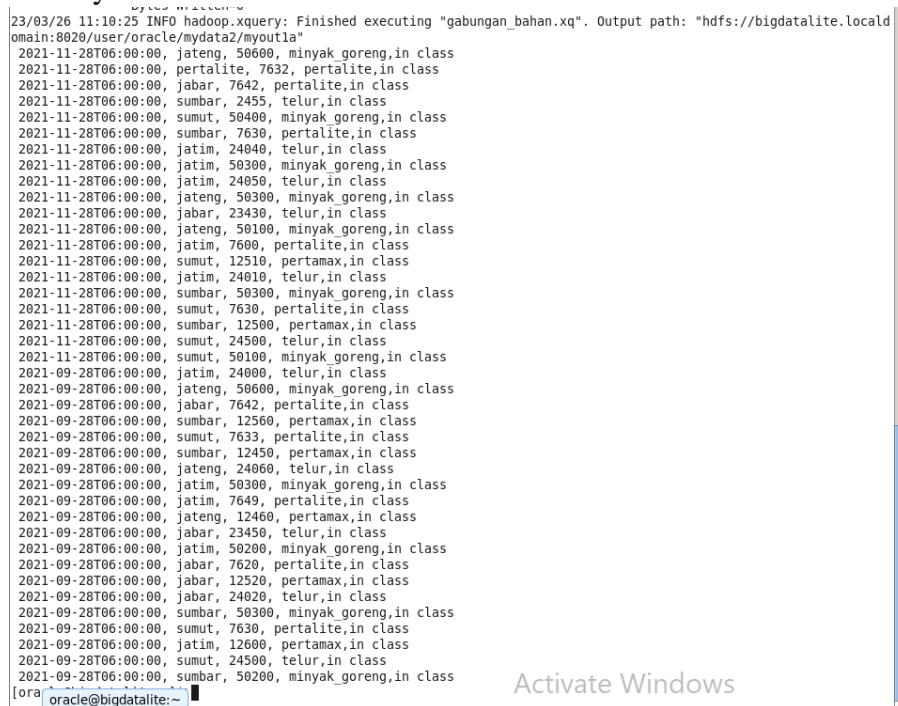


```
oracle@bigdatalite:~
File Edit View Search Terminal Help
GNU nano 2.0.9 File: gabungan_bahan.xq Modified
import module "oxh:text";
for $line in text:collection ("bahanpokok*.txt")
return text:put($line || ",in class")
```

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.locald
omain: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
```

■ 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))
```

■ Inerjoin2.xq

```
GNU nano 2.0.9 File: innerjoin2.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

- Innerjoin1.xq

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar innerjoin1.xq -output ./mydata2/myoutinnerjoin1 -print
```

- Innerjoin2.xq

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar innerjoin2.xq -output ./mydata2/myoutinnerjoin2 -print
```

Hasilnya:

- Innerjoin1.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
```

- Innerjoin2.xq

```
23/03/26 11:56:34 INFO hadoop.xquery: Finished executing "innerjoin2.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata2/myoutinnerjoin2"
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.localdomain:8020/user/oracle/mydata2/myoutouterjoinn"
jabar 7
jateng 6
jatim 10
sumbar 8
sumut 8
[oracle@bigdatalite ~]$
```

▪ Semijoins

1. Membuat file bernama “semjioinn.xq”

```
GNU nano 2.0.9 File: semjioinn.xq Modified
import module "oxh:text";
for $userLine in text:collection("mydata2/provinsi.txt")
let $userId := fn:tokenize($userLine, "\s*:\s*")[1]

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

return text:put($userId)
```

2. Menjalankan Xquery dengan syntax

```
[oracle@bigdatalite ~]$ hadoop jar $OXH_HOME/lib/oxh.jar semjioinn.xq -output ./mydata2/myoutsemjioinn -print
23/03/26 12:28:53 INFO hadoop.xquery: Finished executing "semjioinn.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata2/myoutsemjioinn"
jabar
jateng
jatim
sumbar
sumut
[oracle@bigdatalite ~]$
```

▪ Multiple Outputs

1. Membuat file bernama “multi.xq”

```
GNU nano 2.0.9 File: multiplejoin.xq Modified
import module "oxh:text";
for $visitLine in text:collection("mydata2/bahanpokok*.log")
let $visitCode := xs:integer(fn:tokenize($visitLine, "\s*:\s*")[3])
return if ($visitCode eq 401) then text:trace($visitLine) else text:put($visitLine)
```

2. Menjalankan Xquery dengan syntax

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

```
Hasilnya:
23/03/26 12:36:54 INFO hadoop.xquery: Finished executing "multiplejoin.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/mydata2/myoutmultiplejoin"
2021-11-28T06:00:00, jateng, 50600, minyak goreng
2021-11-28T06:00:00, pertalite, 7632, pertalite
2021-11-28T06:00:00, jabar, 7642, pertalite
2021-11-28T06:00:00, sumbar, 2455, telur
2021-11-28T06:00:00, sumut, 50400, minyak goreng
2021-11-28T06:00:00, sumbar, 7630, pertalite
2021-11-28T06:00:00, jatim, 24040, telur
2021-11-28T06:00:00, jatim, 50300, minyak goreng
2021-11-28T06:00:00, jatim, 24050, telur
2021-11-28T06:00:00, jateng, 50300, minyak goreng
2021-11-28T06:00:00, jabar, 23430, telur
2021-11-28T06:00:00, jateng, 50100, 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, 50300, minyak goreng
2021-11-28T06:00:00, sumut, 7630, pertalite
2021-11-28T06:00:00, sumbar, 12500, pertamax
2021-11-28T06:00:00, sumut, 24500, telur
2021-11-28T06:00:00, sumut, 50100, minyak goreng
2021-09-28T06:00:00, jatim, 24000, telur
2021-09-28T06:00:00, jateng, 50600, minyak goreng
2021-09-28T06:00:00, jabar, 7642, pertalite
2021-09-28T06:00:00, sumbar, 12560, pertamax
2021-09-28T06:00:00, sumut, 7633, pertalite
2021-09-28T06:00:00, sumbar, 12450, pertamax
2021-09-28T06:00:00, jateng, 24060, telur
2021-09-28T06:00:00, jatim, 50300, minyak goreng
2021-09-28T06:00:00, jatim, 7649, pertalite
2021-09-28T06:00:00, jateng, 12460, pertamax
2021-09-28T06:00:00, jabar, 23450, telur
2021-09-28T06:00:00, jatim, 50200, minyak goreng
2021-09-28T06:00:00, jabar, 7620, pertalite
2021-09-28T06:00:00, jabar, 12520, pertamax
2021-09-28T06:00:00, jabar, 24020, telur
2021-09-28T06:00:00, sumbar, 50300, minyak goreng
2021-09-28T06:00:00, sumut, 7630, pertalite
2021-09-28T06:00:00, jatim, 12600, pertamax
2021-09-28T06:00:00, sumut, 24500, telur
2021-09-28T06:00:00, sumbar, 50200, minyak goreng
[oracle@bigdatalite ~]$
```

- **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

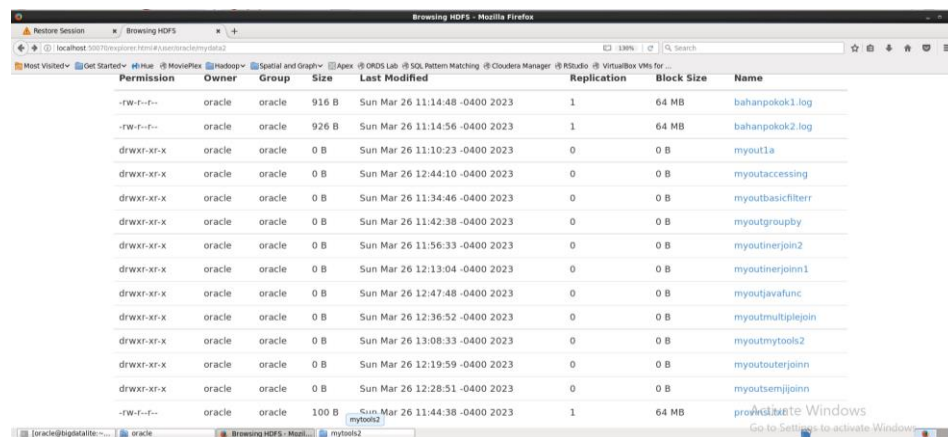
```
[oracle@bigdatalite mytools2]$ hadoop jar $OXH_HOME/lib/oxh.jar -files tools1.xq tools2.xq -output ./mydata2/myoutmyt  
ools2 -print
```

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]$
```

Activate Windows

Secara keseluruhan hasil diatas dapat dilihat pada direktori hdfs



Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	oracle	oracle	916 B	Sun Mar 26 11:14:48 -0400 2023	1	64 MB	bahanpokok1.log
-rw-r--r--	oracle	oracle	926 B	Sun Mar 26 11:14:56 -0400 2023	1	64 MB	bahanpokok2.log
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 11:10:23 -0400 2023	0	0 B	myout1a
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 12:44:10 -0400 2023	0	0 B	myoutaccessing
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 11:34:46 -0400 2023	0	0 B	myoutbasicfilter
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 11:42:38 -0400 2023	0	0 B	myoutgroupby
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 11:56:33 -0400 2023	0	0 B	myoutinnerjoin2
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 12:13:04 -0400 2023	0	0 B	myoutinnerjoin1
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 12:47:48 -0400 2023	0	0 B	myoutjavafunc
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 12:36:52 -0400 2023	0	0 B	myoutmultiplejoin
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 13:08:33 -0400 2023	0	0 B	myoutmytools2
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 12:19:59 -0400 2023	0	0 B	myoutouterjoinn
drwxr-xr-x	oracle	oracle	0 B	Sun Mar 26 12:28:51 -0400 2023	0	0 B	myoutsemjoinn
-rw-r--r--	oracle	oracle	100 B	Sun Mar 26 11:44:38 -0400 2023	1	64 MB	provinsi