

LAPORAN
RENCANA TUGAS MANDIRI (RTM) Ke-4
MATA KULIAH BIG DATA
“ANALISIS STATISTIKA DESKRIPTIF PADA
DATA DENGAN HIVE DAN XQUERY”



DISUSUN OLEH:

Citra Amelia Intan Permadani (21083010004)

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

Analisis Statistika Deskriptif pada Data Suhu di Hive

Langkah awal:

- Buatlah direktori atau folder baru terlebih dahulu, disini saya membuat folder dengan nama *noa_coba* dengan perintah *mkdir*. (`mkdir noa_coba`)
- Masuk ke dalam direktori *noa_coba* dengan perintah *cd* lalu enter dan buka hive.

```
[oracle@bigdatalite ~]$ cd noa_coba
[oracle@bigdatalite noa_coba]$ hive
```

```
Logging initialized using configuration in jar:file:/usr/lib/hive/lib/hive-commo
n-1.1.0-cdh5.13.1.jar!/hive-log4j.properties
WARNING: Hive CLI is deprecated and migration to Beeline is recommended.
hive> █
```

- Buat tabel suhu menggunakan perintah *create* yang terdiri dari kolom tahun, suhu, dan integer dengan keterangan data berisi integer.

```
|hive> create table suhu(tahun int, suhu int, kualitas int);█
```

- Buat file baru didalam hadoop yang nantinya digunakan untuk menyimpan data suhu dengan nama *data_suhu_hive* menggunakan perintah *hadoop fs -mkdir*

```
hadoop fs -mkdir data_suhu_hive█
```

- Buat external tabel dari data suhu sehingga dapat diakses dan diolah menggunakan hive dengan nama *suhutemp* berisi kolom tahun, suhu, dan kualitas yang bertipe data integer. Kemudian untuk mengatur format pembatasan antar kolom menggunakan karakter tab (`\t`) sebagai pemisah dan script `STORED AS TEXTFILE` berarti file data eksternal berupa file teks biasa. Terakhir `LOCATION` menunjukkan lokasi file data eksternal dalam direktori.

```
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';█
```

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 dikelompokkan berdasarkan masing-masing tahun.

- Nilai Suhu Maksimum

- Dengan fungsi agregasi MAX digunakan untuk memilih (*select*) kolom tahun dan suhu serta memberikan nama *max_suhu* pada hasil penghitungan nilai maksimum suhu.
- *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
- *group by tahun* digunakan mengelompokkan data berdasarkan kolom tahun untuk menghitung nilai maksimum suhu dengan nilai tahun yang sama.
- Perhitungan yang dimaksud adalah query akan menghitung nilai maksimum suhu untuk setiap kelompok data dengan nilai yang sama pada kolom tahun. Jadi, nilai suhu yang memiliki tahun yang sama akan dikelompokkan lalu diambil nilai maksimumnya.

```
oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
Time taken: 0.168 seconds, Fetched: 803975 row(s)
hive> SELECT tahun, MAX(suhu) AS max_suhu
> FROM suhutemp
> GROUP BY tahun;
Query ID = oracle_20230316082525_3281a29d-e686-4f0a-88d4-f8258f0e9f32
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1678967293930_0001, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1678967293930_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678967293930_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-16 08:25:40,987 Stage-1 map = 0%, reduce = 0%
2023-03-16 08:25:47,725 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.82 se
c
2023-03-16 08:25:54,042 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.22
sec
MapReduce Total cumulative CPU time: 3 seconds 220 msec
Ended Job = job_1678967293930_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.22 sec HDFS Read: 8851398
HDFS Write: 288 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 220 msec
OK
1901 999
1902 328
1903 999

Hasil:
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
1924 456
1925 378
1926 999
1927 999
1928 999
1929 999
1930 999
1931 999
1932 999
Time taken: 28.079 seconds, Fetched: 32 row(s)
hive>
```

- Nilai Suhu Minimum
 - Dengan fungsi agregasi MIN digunakan untuk memilih (*select*) kolom tahun dan suhu serta memberikan nama *min_suhu* pada hasil penghitungan nilai minimum suhu.
 - *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
 - *group by tahun* digunakan mengelompokkan data berdasarkan kolom tahun untuk menghitung nilai minimum suhu dengan nilai tahun yang sama.
 - Perhitungan yang dimaksud adalah query akan menghitung nilai minimum suhu untuk setiap kelompok data dengan nilai yang sama pada kolom tahun. Jadi, nilai suhu yang memiliki tahun yang sama akan dikelompokkan lalu diambil nilai minimumnya.

```

oracle@bigdatalite:~/noa_col
File Edit View Search Terminal Help
Time taken: 28.079 seconds, Fetched: 32 row(s)
hive> SELECT tahun, MIN(suhu) AS min_suhu
> FROM suhutemp
> GROUP BY tahun;
Query ID = oracle_20230316082929_cc67d6dc-8754-47ae-9cbf-a2d2ecc022a5
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1678967293930_0002, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1678967293930_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678967293930_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-16 08:29:20,256 Stage-1 map = 0%, reduce = 0%
2023-03-16 08:29:26,565 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.64 se
c
2023-03-16 08:29:31,812 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.58
sec
MapReduce Total cumulative CPU time: 3 seconds 580 msec
Ended Job = job_1678967293930_0002
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.58 sec HDFS Read: 8851429
HDFS Write: 224 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 580 msec
OK
1901 0
1902 0
1903 0

```

Hasil:

```

OK
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
1925 0
1926 0
1927 0
1928 0
1929 0
1930 0
1931 0
1932 0
Time taken: 23.136 seconds, Fetched: 32 row(s)
hive> █

```

- Nilai Rata-Rata
 - Dengan fungsi agregasi AVG digunakan untuk memilih (*select*) kolom tahun dan suhu serta memberikan nama *rata_suhu* pada hasil penghitungan nilai rata-rata suhu.
 - *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
 - *group by tahun* digunakan mengelompokkan data berdasarkan kolom tahun untuk menghitung nilai rata-rata suhu dengan nilai tahun yang sama.
 - Perhitungan yang dimaksud adalah query akan menghitung nilai rata-rata suhu untuk setiap kelompok data dengan nilai yang sama pada kolom tahun. Jadi, nilai suhu yang memiliki tahun yang sama akan dikelompokkan lalu diambil nilai rata-ratanya.

```

oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
hive> SELECT tahun, AVG(suhu) AS rata_suhu
> FROM suhutemp
> GROUP BY tahun;
Query ID = oracle_20230316083434_f99861ad-8104-420a-83bb-ee4af8a64acc
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1678967293930_0003, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1678967293930_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678967293930_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-16 08:34:23,709 Stage-1 map = 0%, reduce = 0%
2023-03-16 08:34:29,986 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.74 sec
2023-03-16 08:34:37,315 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.68 sec
MapReduce Total cumulative CPU time: 3 seconds 680 msec
Ended Job = job_1678967293930_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.68 sec HDFS Read: 8851880 HDFS Write: 737 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 680 msec
OK
1901 93.8994668697639
1902 73.78385376999239
1903 78.52105584375954

```

Hasil:

```

OK
1901 93.8994668697639
1902 73.78385376999239
1903 78.52105584375954
1904 74.4364934670313
1905 75.15333028501753
1906 83.6172016952068
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: 28.58 seconds, Fetched: 32 row(s)
hive>

```

- Nilai Varians

- Dengan fungsi agregasi VAR_POP digunakan untuk memilih (*select*) kolom tahun dan suhu serta memberikan nama *varians_suhu* pada hasil penghitungan nilai variansi dari populasi suhu.
- *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
- *group by tahun* digunakan mengelompokkan data berdasarkan kolom tahun untuk menghitung nilai varians suhu dengan nilai tahun yang sama.
- Perhitungan yang dimaksud adalah query akan menghitung nilai variansi dari populasi suhu untuk setiap kelompok data dengan nilai yang sama pada kolom tahun. Jadi, nilai suhu yang memiliki tahun yang sama akan dikelompokkan lalu diambil nilai variansinya.

```

oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
hive> SELECT tahun, VAR_POP(suhu) AS varians_suhu
> FROM suhutemp
> GROUP BY tahun;
Query ID = oracle_20230316083636_92362a60-bce4-4c4b-bda2-3968f2f169f8
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1678967293930_0004, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1678967293930_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678967293930_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-16 08:36:17,383 Stage-1 map = 0%, reduce = 0%
2023-03-16 08:36:22,935 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.83 sec
2023-03-16 08:36:29,206 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.41 sec
MapReduce Total cumulative CPU time: 3 seconds 410 msec
Ended Job = job_1678967293930_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.41 sec HDFS Read: 8851910 HDFS Write: 735 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 410 msec
OK
1901 4763.648994384471
1902 2952.782069839996
1903 9037.802806575155

```

Hasil:

```

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: 26.555 seconds, Fetched: 32 row(s)
hive>

```

- Nilai Standar Deviasi

- Dengan fungsi agregasi STDDEV_POP digunakan untuk memilih (*select*) kolom tahun dan suhu serta memberikan nama *std_deviasi_suhu* pada hasil penghitungan nilai standar deviasi populasi suhu.
- *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
- *group by tahun* digunakan mengelompokkan data berdasarkan kolom tahun untuk menghitung nilai standar deviasi populasi suhu dengan nilai tahun yang sama.
- Perhitungan yang dimaksud adalah query akan menghitung nilai standar deviasi dari populasi suhu untuk setiap kelompok data dengan nilai yang sama pada kolom tahun. Jadi, nilai suhu yang memiliki tahun yang sama akan dikelompokkan lalu diambil nilai standar deviasinya.

```
oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
hive> SELECT tahun, STDDEV_POP(suhu) AS std_deviasi_suhu
> FROM suhutemp
> GROUP BY tahun;
Query ID = oracle_20230316083737_0e446d1b-efbb-4ce7-a671-f0fb4bb948aa
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1678967293930_0005, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1678967293930_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678967293930_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-16 08:38:07,884 Stage-1 map = 0%, reduce = 0%
2023-03-16 08:38:13,093 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.64 sec
2023-03-16 08:38:19,382 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.04 sec
MapReduce Total cumulative CPU time: 3 seconds 40 msec
Ended Job = job_1678967293930_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 3.04 sec HDFS Read: 8851905 HDFS Write: 743 SUCCESS
Total MapReduce CPU Time Spent: 3 seconds 40 msec
OK
1901 69.01919294214089
1902 54.339507449368696
1903 95.06735931209595
```

Hasil:

```
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
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: 22.298 seconds, Fetched: 32 row(s)
hive>
```

- Nilai Percentil

- Dengan fungsi agregasi *percentile* digunakan untuk memilih (*select*) kolom tahun dan suhu serta memberikan nama *persentil_ukuranpersentil* dimulai dari 0.25, 0.50, dan 0.75 pada hasil penghitungan nilai percentil suhu.
- *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
- *group by tahun* digunakan mengelompokkan data berdasarkan kolom tahun untuk menghitung nilai persentil suhu tiap ukuran persentil dengan nilai tahun yang sama.
- Perhitungan yang dimaksud adalah query akan menghitung nilai persentil tiap ukuran persentil suhu untuk setiap kelompok data dengan nilai yang sama pada kolom tahun. Jadi, nilai suhu yang memiliki tahun yang sama akan dikelompokkan lalu diambil nilai persentilnya. Sehingga akan menampilkan output 4 kolom dengan keterangan kolom 1: tahun, kolom 2: persentil ukuran 0.25, kolom 3: persentil ukuran 0.50, dan kolom 4: persentil ukuran 0.75

```
oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
hive> SELECT tahun, percentile(suhu, 0.25) AS persentil_25, percentile(suhu, 0.5) AS persentil_50, percentile(suhu, 0.75) AS persentil_75
> FROM suhutemp
> GROUP BY tahun;
Query ID = oracle_20230316083939_12458f03-4583-47d8-a2eb-b77c09dfe238
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1678967293930_0006, Tracking URL = http://bigdatalite.localdomain:8088/proxy/application_1678967293930_0006/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1678967293930_0006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-16 08:39:58,025 Stage-1 map = 0%, reduce = 0%
2023-03-16 08:40:04,322 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.02 sec
2023-03-16 08:40:12,709 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.17 sec
MapReduce Total cumulative CPU time: 5 seconds 170 msec
Ended Job = job_1678967293930_0006
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.17 sec HDFS Read: 8854077 HDFS Write: 675 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 170 msec
OK
1901 33.0 89.0 144.0
1902 28.0 67.0 111.0
1903 22.0 56.0 122.0
```

Hasil:

```
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: 29.688 seconds, Fetched: 32 row(s)
hive>
```


- b. Persentase perubahan rata-rata suhu di antara 2 tahun, misalnya antara tahun 1902-1903
- *select* digunakan memilih data tahun yang akan dihitung dari tabel *suhutemp*
 - Lalu menghitung rata-rata suhu pada tahun 1902 dan 1903 dengan fungsi *AVG* dan menghitung selisih rata-rata diantara kedua suhu yang kemudian dibagi dengan rata-rata suhu di tahun 1902.
 - **100* digunakan untuk menampilkan hasil perhitungan dalam bentuk persentase.
 - *as percentage_change* digunakan memberikan nama pada hasil perhitungan.
 - *from.....where...in...* digunakan untuk memilih data dari tabel *suhutemp* dengan kondisi berdasarkan data di tahun 1902 dan 1903.
 - Dan didapatkan hasil perubahan rata-rata suhu diantara tahun 1902-1903 adalah 6.42%

```

oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
hive> SELECT
> ((AVG(CASE WHEN tahun = 1903 THEN suhu END) - AVG(CASE WHEN tahun = 19
02 THEN suhu END)) / AVG(CASE WHEN tahun = 1902 THEN suhu END)) * 100 AS percent
age_change
> FROM
> suhutemp
> WHERE
> tahun IN (1902, 1903);
Query ID = oracle_20230320214646_bef5577d-5498-459e-860c-5274bdc8317e
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_1679360697203_0001, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1679360697203_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1679360697203_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-20 21:46:39,891 Stage-1 map = 0%, reduce = 0%
2023-03-20 21:46:47,453 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.5 sec
2023-03-20 21:46:53,824 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.73
sec
MapReduce Total cumulative CPU time: 4 seconds 730 msec
Ended Job = job_1679360697203_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.73 sec HDFS Read: 8854036
HDFS Write: 18 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 730 msec
OK
6.420377673053657
Time taken: 29.566 seconds, Fetched: 1 row(s)
hive>

```

Disini saya memberikan 2 contoh dengan cara yang sama di tahun 1931 dan 1932. Dan didapatkan hasil perubahan rata-rata suhu diantara tahun 1931-1932 adalah 4.54%

```

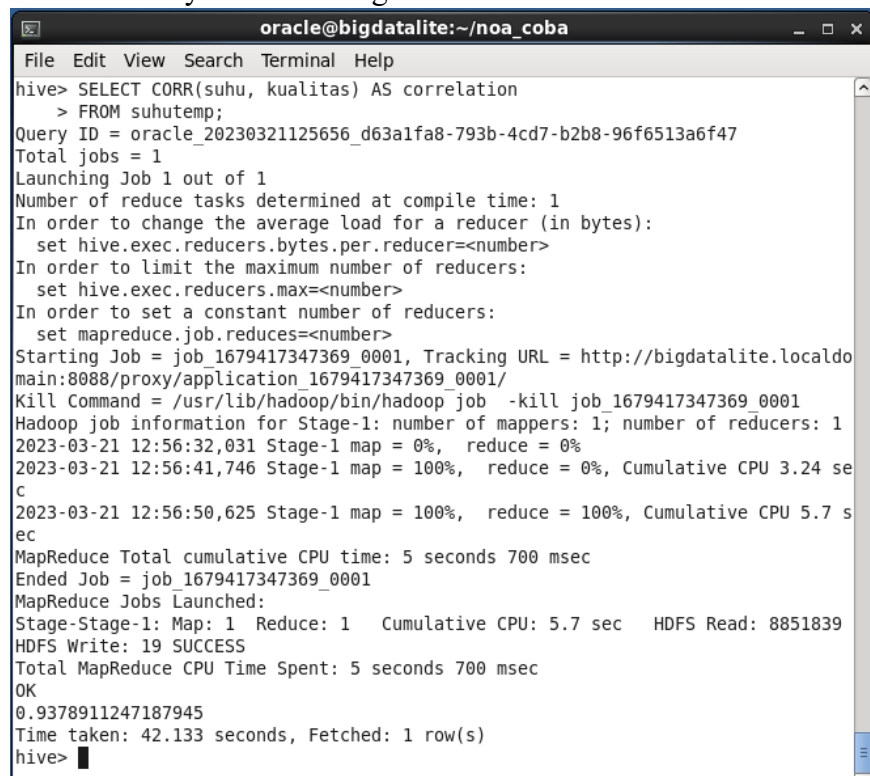
oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
hive> SELECT
> ((AVG(CASE WHEN tahun = 1932 THEN suhu END) - AVG(CASE WHEN tahun = 1931 THEN suhu END
)) / AVG(CASE WHEN tahun = 1931 THEN suhu END)) * 100 AS percentage_change
> FROM
> suhutemp
> WHERE
> tahun IN (1931, 1932);
Query ID = oracle_20230320215353_c91112de-b472-4b12-b37f-2f125abdd4a7
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_1679360697203_0003, Tracking URL = http://bigdatalite.localdomain:8088/proxy/
application_1679360697203_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1679360697203_0003
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-20 21:53:24,448 Stage-1 map = 0%, reduce = 0%
2023-03-20 21:53:32,868 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.76 sec
2023-03-20 21:53:40,232 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.1 sec
MapReduce Total cumulative CPU time: 5 seconds 100 msec
Ended Job = job_1679360697203_0003
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.1 sec HDFS Read: 8854067 HDFS Write: 18 S
UCCESS
Total MapReduce CPU Time Spent: 5 seconds 100 msec
OK
4.543323800019632
Time taken: 28.744 seconds, Fetched: 1 row(s)
hive>

```

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

Pertanyaan:

- Menemukan korelasi antara suhu dan kualitas
 - Dengan fungsi agregasi CORR digunakan untuk memilih (*select*) kolom suhu dan kualitas serta memberikan nama *correlation* pada hasil penghitungan nilai korelasi antara suhu dan kualitas.
 - *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
 - Perhitungan yang dimaksud adalah query akan menghitung nilai korelasi antara tabel suhu dan kualitas.
 - Dan dihasilkan bahwa korelasi antara suhu dan kualitas sebesar 0.94 yang berarti korelasi diantara keduanya sangat kuat positif karena mendekati 1. Sehingga jika salah satu variabel meningkat maka variabel lainnya akan meningkat.



```
oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
hive> SELECT CORR(suhu, kualitas) AS correlation
> FROM suhutemp;
Query ID = oracle_20230321125656_d63a1fa8-793b-4cd7-b2b8-96f6513a6f47
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_1679417347369_0001, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1679417347369_0001/
Kill Command = /usr/lib/hadoop/bin/hadoop_job -kill job_1679417347369_0001
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-21 12:56:32,031 Stage-1 map = 0%, reduce = 0%
2023-03-21 12:56:41,746 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.24 se
c
2023-03-21 12:56:50,625 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 5.7 s
ec
MapReduce Total cumulative CPU time: 5 seconds 700 msec
Ended Job = job_1679417347369_0001
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 5.7 sec HDFS Read: 8851839
HDFS Write: 19 SUCCESS
Total MapReduce CPU Time Spent: 5 seconds 700 msec
OK
0.9378911247187945
Time taken: 42.133 seconds, Fetched: 1 row(s)
hive>
```

- Mendeteksi apakah terdapat missing value
 - Dengan fungsi agregasi COUNT digunakan untuk memilih (*select*) kolom suhu serta memberikan nama *jumlah_missing_value* pada hasil pendeteksian nilai missing value pada kolom suhu.
 - *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
 - *where...is...* digunakan untuk memberikan kondisi dengan kolom suhu yang bernilai null (bukan 0).
 - Dihasilkan bahwa dalam kolom suhu tidak terdapat missing value.

```
oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
Time taken: 0.175 seconds, Fetched: 803975 row(s)
hive> SELECT COUNT(*) AS jumlah_missing_value FROM suhutemp WHERE suhu IS NULL;
Query ID = oracle_20230321132020_f772fa1a-9f57-4416-b26d-ec1f9fe425c0
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_1679417347369_0004, Tracking URL = http://bigdatalite.localdomain:8088/proxy/app
lication 1679417347369_0004/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1679417347369_0004
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-21 13:21:02,122 Stage-1 map = 0%, reduce = 0%
2023-03-21 13:21:09,690 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.13 sec
2023-03-21 13:21:14,898 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.38 sec
MapReduce Total cumulative CPU time: 4 seconds 380 msec
Ended Job = job_1679417347369_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.38 sec HDFS Read: 8851955 HDFS Write: 2 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 380 msec
OK
0
Time taken: 34.094 seconds, Fetched: 1 row(s)
```

- Disini saya juga melakukan deteksi missing value pada kolom kualitas. Dihasilkan bahwa dalam kolom kualitas tidak terdapat missing value.

```
oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
2023-03-21 13:21:14,898 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.38 sec
MapReduce Total cumulative CPU time: 4 seconds 380 msec
Ended Job = job_1679417347369_0004
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.38 sec HDFS Read: 8851955 HDFS Write: 2 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 380 msec
OK
0
Time taken: 34.094 seconds, Fetched: 1 row(s)
hive> SELECT COUNT(*) AS jumlah_missing_value FROM suhutemp WHERE kualitas IS NULL;
Query ID = oracle_20230321132121_163693ae-9e2a-4be7-88d6-196255242f4d
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_1679417347369_0005, Tracking URL = http://bigdatalite.localdomain:8088/proxy/app
lication 1679417347369_0005/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1679417347369_0005
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-21 13:21:54,595 Stage-1 map = 0%, reduce = 0%
2023-03-21 13:22:01,002 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 2.64 sec
2023-03-21 13:22:07,310 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 4.21 sec
MapReduce Total cumulative CPU time: 4 seconds 210 msec
Ended Job = job_1679417347369_0005
MapReduce Jobs Launched:
Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 4.21 sec HDFS Read: 8851955 HDFS Write: 2 SUCCESS
Total MapReduce CPU Time Spent: 4 seconds 210 msec
OK
0
Time taken: 26.775 seconds, Fetched: 1 row(s)
hive>
```

- Menemukan 10 tahun teratas dengan suhu rata-rata tertinggi
 - Dengan fungsi agregasi AVG digunakan untuk memilih (*select*) kolom suhu serta memberikan nama *avg_suhu* pada hasil penghitungan nilai rata-rata suhu.
 - *from* menunjukkan bahwa tabel yang digunakan adalah tabel *suhutemp* yang sebelumnya telah dibuat.
 - *group by tahun* digunakan mengelompokkan data berdasarkan kolom tahun untuk menghitung nilai rata-rata suhu dengan nilai tahun yang sama.
 - *order by* digunakan mengurutkan hasil query berdasarkan nilai rata-rata suhu dari tertinggi ke terendah dengan fungsi *desc*.
 - *limit* berarti membatasi jumlah hasil query yang ditampilkan hanya 10 tahun paling tinggi rata-rata suhunya.

```

oracle@bigdatalite:~/noa_coba
File Edit View Search Terminal Help
Time taken: 42.133 seconds, Fetched: 1 row(s)
hive> SELECT tahun, AVG(suhu) AS avg_suhu
> FROM suhutemp
> GROUP BY tahun
> ORDER BY avg_suhu DESC
> LIMIT 10;
Query ID = oracle_20230321125858_02dd1a4f-250c-4a7c-ba7c-41041aa0c41c
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1679417347369_0002, Tracking URL = http://bigdatalite.localdo
main:8088/proxy/application_1679417347369_0002/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1679417347369_0002
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
2023-03-21 12:58:23,385 Stage-1 map = 0%, reduce = 0%
2023-03-21 12:58:28,718 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 1.81 se
c
2023-03-21 12:58:35,041 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 3.15
sec
MapReduce Total cumulative CPU time: 3 seconds 150 msec
Ended Job = job_1679417347369_0002
Launching Job 2 out of 2
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_1679417347369_0003, Tracking URL = http://bigdatalite.localdomain:8088/proxy/app
lication_1679417347369_0003/
Kill Command = /usr/lib/hadoop/bin/hadoop job -kill job_1679417347369_0003

```

Hasil:

```

Total MapReduce CPU Time Spent: 5 seconds 640 msec
OK
1932    167.94986493146183
1931    160.65097112536066
1930    147.69661222020568
1929    131.9264331407987
1928    117.86021212945336
1927    111.78934446354039
1926    100.65950269853508
1917    94.20697541452259
1901    93.8994668697639
1912    92.43987506507028
Time taken: 48.757 seconds, Fetched: 10 row(s)
hive>

```

Analisis Data Bahan Pokok dan Saham Menggunakan Xquery di Hive

Unduh dataset dummy - Saham dan Harga Sembako lalu analisislah menggunakan bentuk-bentuk Xquery transformation yang sesuai.

<https://drive.google.com/drive/folders/182b5TikHcqCe2vAzgfabaNNjAfG5Qh6s?usp=sharing>

Langkah awal:

- Download terlebih dahulu dataset Saham dan Harga Sembako dari link gdrive di atas. Kemudian buka terminal linux bigdata lite.

- Buat direktori baru dengan perintah *mkdir*, disini saya menamainya *joindata*

```
[oracle@bigdatalite ~]$ mkdir joindata
```

- Copy kan file dataset yang telah didownload ke dalam direktori *joindata*. Daftar file yang di copy adalah log dan txt lalu masuk ke dalam direktori *joindata*

```
[oracle@bigdatalite ~]$ cp trading*.log joindata
[oracle@bigdatalite ~]$ cd joindata
[oracle@bigdatalite joindata]$ ls
trading1.log trading2.log
[oracle@bigdatalite joindata]$ cd
[oracle@bigdatalite ~]$ cp bahanpokok*.log joindata
[oracle@bigdatalite ~]$ cd joindata
[oracle@bigdatalite joindata]$ ls
bahanpokok1.log bahanpokok2.log trading1.log trading2.log
[oracle@bigdatalite joindata]$
```

- Buat direktori dalam hadoop dengan perintah *hadoop fs -mkdir -p /user/oracle/joindata*

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

- Copy kan file log ke dalam hadoop dengan perintah *hdfs dfs -copyFromLocal trading*.log /user/oracle/joindata* untuk file trading dan *hdfs dfs -copyFromLocal bahanpokok*.log /user/oracle/joindata* untuk file bahanpokok

```
[oracle@bigdatalite joindata]$ hdfs dfs -copyFromLocal trading*.log /user/oracle/joindata
[oracle@bigdatalite joindata]$ hdfs dfs -copyFromLocal bahanpokok*.log /user/oracle/joindata
[oracle@bigdatalite joindata]$
```

- Copy juga file txt ke dalam hadoop dengan perintah *hdfs dfs -copyFromLocal provinsi.txt /user/oracle/joindata* untuk file provinsi dan *hdfs dfs -copyFromLocal trader.txt /user/oracle/joindata* untuk file trader.

```
oracle@bigdatalite:~/joindata
File Edit View Search Terminal Help
[oracle@bigdatalite joindata]$ ls
bahanpokok1.log bahanpokok2.log provinsi.txt trader.txt trading1.log trading2.log
[oracle@bigdatalite joindata]$ hdfs dfs -copyFromLocal provinsi.txt /user/oracle/joindata
[oracle@bigdatalite joindata]$ hdfs dfs -copyFromLocal trader.txt /user/oracle/joindata
```

1. Data Harga Sembako

Dengan menggunakan data harga sembako, saya menganalisis menggunakan Xquery Inner Join dan Outer Join.

- Inner Join

- Buat file xq dengan perintah *sudo nano innerjoin5.xq*



- Maka akan terbuka GNU Nano untuk membuat isi file xq diatas. Lalu isikan file innerjoin5.xq seperti di bawah ini. Disini saya menggunakan Xquery inner join di file bahanpokok*.log dan provinsi.txt dengan kondisi menampilkan data dari bahanpokok*log yang harga sembakonya lebih dari 7500.

Prosesnya:

- ✓ Import module *oxh:text*
- ✓ Muat data dari file *provinsi.txt* ke dalam variabel *\$provinsiLine*
- ✓ Memisahkan kolom dari variabel *\$provinsiLine* menjadi 2 bagian berdasarkan tanda titik dua (:) dan menyimpan bagian pertama [1] sebagai variabel *\$provinsiId*.
- ✓ Mengulang lewat tiap baris dari file yang cocok dengan pola *bahanpokok*.log*
- ✓ Memisahkan setiap kolom dari file *bahanpokok*log* berdasarkan tanda koma (,) dan menyimpan bagian kedua [2] sebagai variabel *\$bahanpokokProvinsiId* dan bagian bagian ketiga [3] sebagai variabel *\$bahanpokokHarga* yang dikonversi ke integer.
- ✓ Memfilter baris yang cocok antara *\$bahanpokokProvinsiId* dengan *\$provinsiId* dengan kondisi *\$bahanpokokHarga* lebih besar (*gt*) dari 7500
- ✓ Return atau mengembalikan hasil dari jumlah baris yang memenuhi kondisi untuk tiap *\$page* yang terpisah oleh spasi.

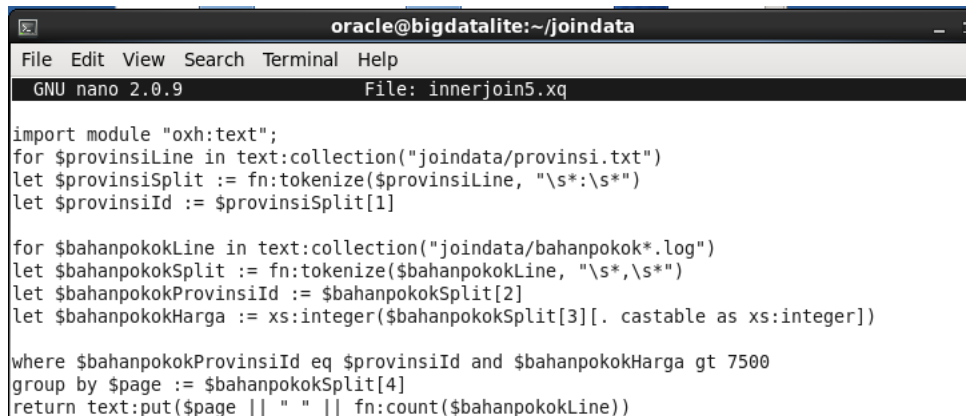
```

import module "oxh:text";
for $provinsiLine in text:collection("joindata/province.txt")
let $provinsiSplit := fn:tokenize($provinsiLine, "s*:s*")
let $provinsild := $provinsiSplit[1]

for $bahanpokokLine in text:collection("joindata/bahanpokok*.log")
let $bahanpokokSplit := fn:tokenize($bahanpokokLine, "s*,s*")
let $bahanpokokProvinsild := $bahanpokokSplit[2]
let $bahanpokokHarga := xs:integer($bahanpokokSplit[3][. castable as
xs:integer])

Where $bahanpokokProvinsild EQ $provinsild and $bahanpokokHarga gt 7500
group by $page := $bahanpokokSplit[4]
return text:put($page || " " || fn:count($bahanpokokLine))

```



```

oracle@bigdatalite:~/joindata
File Edit View Search Terminal Help
GNU nano 2.0.9 File: innerjoin5.xq

import module "oxh:text";
for $provinsiLine in text:collection("joindata/provinsi.txt")
let $provinsiSplit := fn:tokenize($provinsiLine, "\s*:\s*")
let $provinsiId := $provinsiSplit[1]

for $bahanpokokLine in text:collection("joindata/bahanpokok*.log")
let $bahanpokokSplit := fn:tokenize($bahanpokokLine, "\s*,\s*")
let $bahanpokokProvinsiId := $bahanpokokSplit[2]
let $bahanpokokHarga := xs:integer($bahanpokokSplit[3][. castable as xs:integer])

where $bahanpokokProvinsiId eq $provinsiId and $bahanpokokHarga gt 7500
group by $page := $bahanpokokSplit[4]
return text:put($page || " " || fn:count($bahanpokokLine))

```

- Simpan file dengan CTRL+X dan pilih y, maka file akan tersimpan dan kembali ke terminal.
- Eksekusi file `innerjoin5.xq` dengan perintah `hadoop jar $OXH_HOME/lib/oxh.jar innerjoin5.xq -output ./joindata/join5outinner -print`

Dari perintah diatas, Hadoop diarahkan untuk menjalankan file `innerjoin5.xq` dari direktori `joindata` menggunakan OxH (Open Xquery Hadoop) dengan output yang disimpan ke direktori `joindata` dengan nama `join5outinner` lalu untuk mencetak hasilnya di konsol menggunakan opsi `print`. Dan opsi `output` digunakan menunjukkan direktori output dalam Hadoop.


```
[oracle@bigdatalite joindata]$ hadoop jar $OXH_HOME/lib/oxh.jar innerjoin5.xq -output ./joindata/join5outinner -print
23/03/25 03:34:02 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2 @mr2). Copyright (c) 2023, Oracle. All rights reserved.
23/03/25 03:34:02 INFO hadoop.xquery: Executing query "innerjoin5.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/join5outinner"
23/03/25 03:34:04 INFO hadoop.xquery: Submitting map-reduce job "oxh:innerjoin5.xq#0" id="b3a0175b-24d3-495d-94aa-f5713a196d66.0", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/provinsi.txt], output=hdfs://bigdatalite.localdomain:8020/tmp/oxh-oracle/scratch/b3a0175b-24d3-495d-94aa-f5713a196d66.0
23/03/25 03:34:04 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/25 03:34:06 INFO input.FileInputFormat: Total input paths to process : 1
23/03/25 03:34:07 INFO mapreduce.JobSubmitter: number of splits:1
23/03/25 03:34:07 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679726512398_0024
23/03/25 03:34:07 INFO impl.YarnClientImpl: Submitted application application_1679726512398_0024
23/03/25 03:34:07 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_1679726512398_0024/
23/03/25 03:34:07 INFO hadoop.xquery: Submitting map-reduce job "oxh:innerjoin5.xq#1" id="b3a0175b-24d3-495d-94aa-f5713a196d66.1", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/bahanpokok*.log], output=hdfs://bigdatalite.localdomain:8020/tmp/oxh-oracle/scratch/b3a0175b-24d3-495d-94aa-f5713a196d66.1
23/03/25 03:34:07 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/25 03:34:08 INFO input.FileInputFormat: Total input paths to process : 2
23/03/25 03:34:09 INFO mapreduce.JobSubmitter: number of splits:2
23/03/25 03:34:09 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679726512398_0025
23/03/25 03:34:09 INFO impl.YarnClientImpl: Submitted application application_1679726512398_0025
23/03/25 03:34:09 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_1679726512398_0025/
23/03/25 03:34:09 INFO hadoop.xquery: Waiting for map-reduce job oxh:innerjoin5.xq#0
23/03/25 03:34:09 INFO mapreduce.Job: Running job: job_1679726512398_0024
23/03/25 03:34:14 INFO mapreduce.Job: Job job_1679726512398_0024 running in uber mode : false
23/03/25 03:34:14 INFO mapreduce.Job: map 0% reduce 0%
```

- Maka proses map reduce akan berjalan untuk menghitung berapa harga sembako yang lebih dari 7500 beserta keterangan bahan sembakonya. Dan dihasilkan bahwa sembako yang dibeli dengan harga lebih dari 7500 adalah

Minyak goreng – 12

Pertalite – 9

Pertamax – 7

Telur - 10

```
23/03/25 03:36:06 INFO hadoop.xquery: Finished executing "innerjoin5.xq". Output path: "hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/join5outinner"
```

minyak goreng 12

pertalite 9

pertamax 7

telur 10

```
[oracle@bigdatalite joindata]$
```

- Outer Join

- Buat file xq dengan perintah *sudo nano outerjoin5.xq*

```
[oracle@bigdatalite joindata]$ nano outerjoin5.xq
```

- Maka akan terbuka GNU Nano untuk membuat isi file xq diatas. Lalu isikan file outerjoin5.xq seperti di bawah ini.

Disini saya menggunakan Xquery outer join di file bahanpokok*.log dan provinsi.txt dengan kondisi menampilkan data dari bahanpokok*log dimana terdapat baris di kolom provinsi (2) yang tidak sesuai.

Prosesnya:

- ✓ Import module *oxh:text*
- ✓ Looping setiap baris log dari file *bahanpokok*log*
- ✓ Memisahkan setiap baris menggunakan *fn:tokrnize* dan simpan ke dalam variabel *\$bahanpokokSplit* berdasarkan tanda koma (,) dan menyimpan bagian kedua [2] sebagai variabel *\$bahanpokokId*.

- ✓ Mengulang tiap baris dari file *provinsi.txt* dengan *for provinsiLine allowing empty in text:collection("joindata/provinsi.txt")* dimana *allowing empty* digunakan jika tidak ada baris yang cocok dengan kondisi yang ditentukan maka loop akan tetap berjalan.
- ✓ Menyeleksi baris yang memiliki ID provinsi yang sama dengan ID bahan pokok dengan *\$bahanpokokId eq fn:tokenize(., "\s*:\s*")[1]* dengan mengambil baris pertama dari file *provinsi.txt* yang dipisahkan dengan tanda titik koma(:).
- ✓ Grouping hasil seleksi berdasarkan ID bahan pokok dengan *group by*.
- ✓ Return atau mengembalikan hasil dari jumlah baris yang memenuhi kondisi sesuai dengan ID bahan pokok yang sedang diiterasi pada file log dengan menampilkan ID bahan pokok dan jumlah baris pada file txt.

```
import module "oxh:text";

for $bahanpokokLine in
text:collection("joindata/bahanpokok*.log")

let $bahanpokokSplit := fn:tokenize($bahanpokokLine, "\s*:\s*")

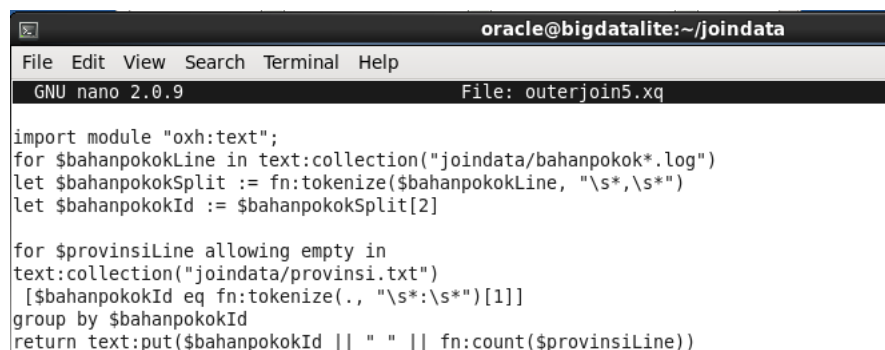
let $bahanpokokId := $bahanpokokSplit[2]


for $provinsiLine allowing empty in
text:collection("joindata/provinsi.txt")

[$bahanpokokId eq fn:tokenize(., "\s*:\s*")[1]]

group by $bahanpokokId

return text:put($bahanpokokId || " " || fn:count($provinsiLine))
```



```
oracle@bigdatalite:~/joindata
File Edit View Search Terminal Help
GNU nano 2.0.9 File: outerjoin5.xq

import module "oxh:text";
for $bahanpokokLine in text:collection("joindata/bahanpokok*.log")
let $bahanpokokSplit := fn:tokenize($bahanpokokLine, "\s*:\s*")
let $bahanpokokId := $bahanpokokSplit[2]

for $provinsiLine allowing empty in
text:collection("joindata/provinsi.txt")
[$bahanpokokId eq fn:tokenize(., "\s*:\s*")[1]]
group by $bahanpokokId
return text:put($bahanpokokId || " " || fn:count($provinsiLine))
```

- Simpan file dengan CTRL+X dan pilih y, maka file akan tersimpan dan kembali ke terminal.

- Eksekusi file `innerjoin5.xq` dengan perintah `hadoop jar $OXH_HOME/lib/oxh.jar outerjoin5.xq -output ./joindata/myouter6join -print`

Dari perintah diatas, Hadoop diarahkan untuk menjalankan file `outerjoin5.xq` dari direktori `joindata` menggunakan OxH (Open Xquery Hadoop) dengan output yang disimpan ke direktori `joindata` dengan nama `myouter6join` lalu untuk mencetak hasilnya di konsol menggunakan opsi `print`. Dan opsi `output` digunakan menunjukkan direktori output dalam Hadoop.

```
[oracle@bigdatalite joindata]$ hadoop jar $OXH_HOME/lib/oxh.jar outerjoin5.xq -output ./joindata/myouter6join -print
23/03/25 04:55:26 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2). Copyright (c)
2023, Oracle. All rights reserved.
23/03/25 04:55:26 INFO hadoop.xquery: Executing query "outerjoin5.xq". Output path: "hdfs://bigdatalite.localdomain:8020
/user/oracle/joindata/myouter6join"
23/03/25 04:55:28 INFO hadoop.xquery: Submitting map-reduce job "oxh:outerjoin5.xq#0" id="804ed380-66c0-4703-b46e-00cca9
f4caf6.0", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/bahanpokok*.log], output=hdfs://bigdatalite.
localdomain:8020/tmp/oxh-oracle/scratch/804ed380-66c0-4703-b46e-00cca9f4caf6.0
23/03/25 04:55:28 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/25 04:55:30 INFO input.FileInputFormat: Total input paths to process : 2
23/03/25 04:55:30 INFO mapreduce.JobSubmitter: number of splits:2
23/03/25 04:55:30 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679726512398_0056
23/03/25 04:55:30 INFO impl.YarnClientImpl: Submitted application application_1679726512398_0056
23/03/25 04:55:31 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_16
79726512398_0056/
23/03/25 04:55:31 INFO hadoop.xquery: Submitting map-reduce job "oxh:outerjoin5.xq#1" id="804ed380-66c0-4703-b46e-00cca9
f4caf6.1", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/provinsi.txt], output=hdfs://bigdatalite.loc
aldomain:8020/tmp/oxh-oracle/scratch/804ed380-66c0-4703-b46e-00cca9f4caf6.1
23/03/25 04:55:31 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/25 04:55:31 INFO input.FileInputFormat: Total input paths to process : 1
23/03/25 04:55:31 INFO mapreduce.JobSubmitter: number of splits:1
23/03/25 04:55:31 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679726512398_0057
23/03/25 04:55:31 INFO impl.YarnClientImpl: Submitted application application_1679726512398_0057
23/03/25 04:55:31 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_16
79726512398_0057/
23/03/25 04:55:31 INFO hadoop.xquery: Waiting for map-reduce job oxh:outerjoin5.xq#0
23/03/25 04:55:31 INFO mapreduce.Job: Running job: job_1679726512398_0056
23/03/25 04:55:37 INFO mapreduce.Job: Job job_1679726512398_0056 running in uber mode : false
23/03/25 04:55:37 INFO mapreduce.Job: map 0% reduce 0%
23/03/25 04:55:47 INFO mapreduce.Job: map 50% reduce 0%
23/03/25 04:55:49 INFO mapreduce.Job: map 100% reduce 0%
23/03/25 04:55:49 INFO mapreduce.Job: Job job_1679726512398_0056 completed successfully
23/03/25 04:55:49 INFO mapreduce.Job: Counters: 31
File System Counters
FILE: Number of bytes read=0
FILE: Number of bytes written=320792
FILE: Number of read operations=0
FILE: Number of large read operations=0
```

- Maka proses map reduce akan berjalan untuk menghitung kata dalam kolom provinsi (2) berdasarkan file provinsi.txt beserta keterangan jumlah kemunculannya. Dan dihasilkan bahwa terdapat satu baris di kolom provinsi (2) dalam file log yang tidak sesuai dengan file provinsi

jabar – 7

jateng – 6

jatim – 10

pertalite – 0

sumbar – 8

sumut - 8

```
23/03/25 04:56:54 INFO hadoop.xquery: Finished executing "outerjoin5.xq". Output path: "hdfs://bigdatalite.localdomain:8
020/user/oracle/joindata/myouter6join"
jabar 7
jateng 6
jatim 10
pertalite 0
sumbar 8
sumut 8
[oracle@bigdatalite joindata]$
```

2. Data Saham

Dengan menggunakan data saham, saya menganalisis menggunakan Xquery Inner Join.

- Inner Join

- Buat file xq dengan perintah *sudo nano innerjoinn1.xq*

```
[oracle@bigdatalite joindata]$ sudo nano innerjoinn1.xq
```

- Maka akan terbuka GNU Nano untuk membuat isi file xq diatas. Lalu isikan file innerjoinn1.xq seperti di bawah ini.

Disini saya menggunakan Xquery inner join di file trading*.log dan trader.txt dengan kondisi menampilkan data dari trading*log dimana saham yang diberikan lebih dari 70.

Prosesnya:

- ✓ Import module *oxh:text*
- ✓ Muat data dari file *trader.txt* ke dalam variabel *\$traderLine*
- ✓ Memisahkan kolom dari variabel *\$traderLine* menjadi 2 bagian berdasarkan tanda titik dua (:) dan menyimpan bagian pertama [1] sebagai variabel *\$traderId*.
- ✓ Mengulang lewat tiap baris dari file yang cocok dengan pola *trading*.log*
- ✓ Memisahkan setiap kolom dari file *trading*log* berdasarkan tanda koma (,) dan menyimpan bagian kedua [2] sebagai variabel *\$tradingTraderId* dan bagian bagian ketiga [3] sebagai variabel *\$tradingSaham* yang dikonversi ke integer.
- ✓ Untuk menampilkan pilihan-pilihan kolom kita dapat menggunakan perintah *concat*. Disini buat variabel *\$tradingPage* untuk menyimpan perintah dalam menampilkan kolom yang diinginkan dalam output yakni kolom kedua dan keempat yang dipisahkan dengan '- '.
- ✓ Memfilter baris yang cocok antara *\$tradingTraderId* dengan *\$traderId* dengan kondisi *\$tradingSaham* lebih besar (*gt*) dari 70
- ✓ Return atau mengembalikan hasil dari jumlah baris yang memenuhi kondisi untuk tiap *\$tradingPage* yang terpisah oleh spasi.

```

import module "oxh:text";

for $traderLine in text:collection("joindata/trader.txt")
let $traderSplit := fn:tokenize($traderLine, "s*:s*")
let $traderId := $traderSplit[1]

for $tradingLine in text:collection("joindata/trading*.log")
let $tradingSplit := fn:tokenize($tradingLine, "s*,s*")
let $tradingTraderId := $traderSplit[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))

```

```

oracle@bigdatalite:~/joindata
File Edit View Search Terminal Help
GNU nano 2.0.9 File: innerjoinn1.xq

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

for $tradingLine in text:collection("joindata/trading*.log")
let $tradingSplit := fn:tokenize($tradingLine, "s*,s*")
let $tradingTraderId := $traderSplit[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))

```

- Simpan file dengan CTRL+X dan pilih y, maka file akan tersimpan dan kembali ke terminal.
- Eksekusi file innerjoinn1.xq dengan perintah *hadoop jar \$OXH_HOME/lib/oxh.jar innerjoinn1.xq -output ./joindata/joinnoutinner -print*

Dari perintah diatas, Hadoop diarahkan untuk menjalankan file *innerjoinn1.xq* dari direktori *joindata* menggunakan OxH (Open Xquery Hadoop) dengan output yang disimpan ke direktori *joindata* dengan nama *joinnoutinner* lalu untuk mencetak hasilnya di konsol menggunakan opsi *print*. Dan opsi *output* digunakan menunjukkan direktori output dalam Hadoop.

```
oracle@bigdatalite:~/joindata
File Edit View Search Terminal Help
[oracle@bigdatalite joindata]$ hadoop jar $OXH_HOME/lib/oxh.jar innerjoinn1.xq -output ./joindata/joinnoutinner -print
23/03/25 03:47:14 INFO hadoop.xquery: OXH: Oracle XQuery for Hadoop 4.9.1 (build 4.9.1-cdh5.0.0-mr2). Copyright (c)
2023, Oracle. All rights reserved.
23/03/25 03:47:15 INFO hadoop.xquery: Executing query "innerjoinn1.xq". Output path: "hdfs://bigdatalite.localdomain:802
0/user/oracle/joindata/joinnoutinner"
23/03/25 03:47:16 INFO hadoop.xquery: Submitting map-reduce job "oxh:innerjoinn1.xq#0" id="d301726c-4348-4af0-86e0-97a1e
e54e994.0", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/trader.txt], output=hdfs://bigdatalite.loca
ldomain:8020/tmp/oxh-oracle/scratch/d301726c-4348-4af0-86e0-97a1ee54e994.0
23/03/25 03:47:16 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/25 03:47:17 WARN hdfs.DFSClient: Caught exception
java.lang.InterruptedException
    at java.lang.Object.wait(Native Method)
    at java.lang.Thread.join(Thread.java:1252)
    at java.lang.Thread.join(Thread.java:1326)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.closeResponder(DFSOutputStream.java:967)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.endBlock(DFSOutputStream.java:785)
    at org.apache.hadoop.hdfs.DFSOutputStream$DataStreamer.run(DFSOutputStream.java:894)
23/03/25 03:47:18 INFO input.FileInputFormat: Total input paths to process : 1
23/03/25 03:47:18 INFO mapreduce.JobSubmitter: number of splits:1
23/03/25 03:47:18 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679726512398_0028
23/03/25 03:47:19 INFO impl.YarnClientImpl: Submitted application application_1679726512398_0028
23/03/25 03:47:19 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_16
79726512398_0028/
23/03/25 03:47:19 INFO hadoop.xquery: Submitting map-reduce job "oxh:innerjoinn1.xq#1" id="d301726c-4348-4af0-86e0-97a1e
e54e994.1", inputs=[hdfs://bigdatalite.localdomain:8020/user/oracle/joindata/trading*.log], output=hdfs://bigdatalite.lo
caldomain:8020/tmp/oxh-oracle/scratch/d301726c-4348-4af0-86e0-97a1ee54e994.1
23/03/25 03:47:19 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/25 03:47:19 INFO input.FileInputFormat: Total input paths to process : 2
23/03/25 03:47:20 INFO mapreduce.JobSubmitter: number of splits:2
23/03/25 03:47:20 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1679726512398_0029
23/03/25 03:47:20 INFO impl.YarnClientImpl: Submitted application application_1679726512398_0029
23/03/25 03:47:20 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_16
79726512398_0029/
23/03/25 03:47:20 INFO hadoop.xquery: Waiting for map-reduce job oxh:innerjoinn1.xq#0
23/03/25 03:47:20 INFO mapreduce.Job: Running job: job_1679726512398_0028
23/03/25 03:47:26 INFO mapreduce.Job: Job job_1679726512398_0028 running in uber mode : false
23/03/25 03:47:26 INFO mapreduce.Job: map 0% reduce 0%
23/03/25 03:47:36 INFO mapreduce.Job: map 100% reduce 0%
23/03/25 03:47:37 INFO mapreduce.Job: Job job_1679726512398_0028 completed successfully
23/03/25 03:47:38 INFO mapreduce.Job: Counters: 30
```

- Maka proses map reduce akan berjalan untuk menghitung harga saham yang diberikan lebih dari 70 beserta keterangannya pemilik saham. Dan dihasilkan bahwa saham yang diberikan dengan harga lebih dari 70 adalah

```
23/03/25 03:48:37 INFO hadoop.xquery: Finished executing "innerjoinn1.xq". Output path: "hdfs://bigdatalite.localdomain:
8020/user/oracle/joindata/joinnoutinner"
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 joindata]$
```