

**LAPORAN**  
**RENCANA TUGAS MANDIRI (RTM) Ke-3**  
**MATA KULIAH BIG DATA B**  
**“Mendistribusikan data teks berita ke dalam HDFS dan  
membuat model pemrograman MapReduce untuk program  
WordCount”**



**DISUSUN OLEH:**

Aulia Nur Fitriani      (21083010051)

**DOSEN PENGAMPU:**

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

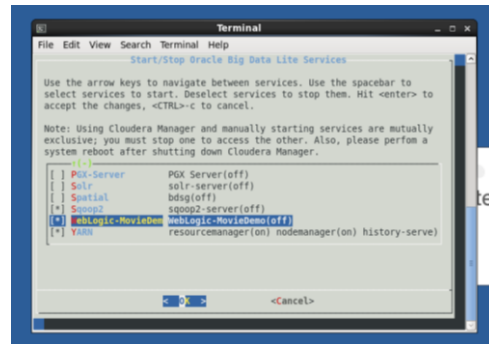
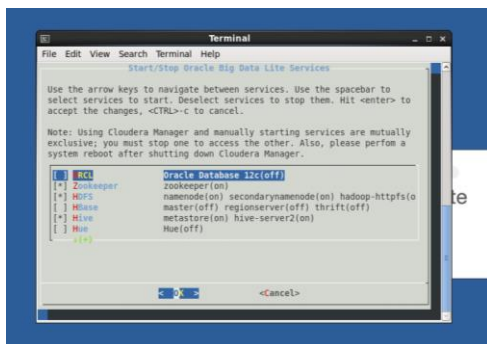
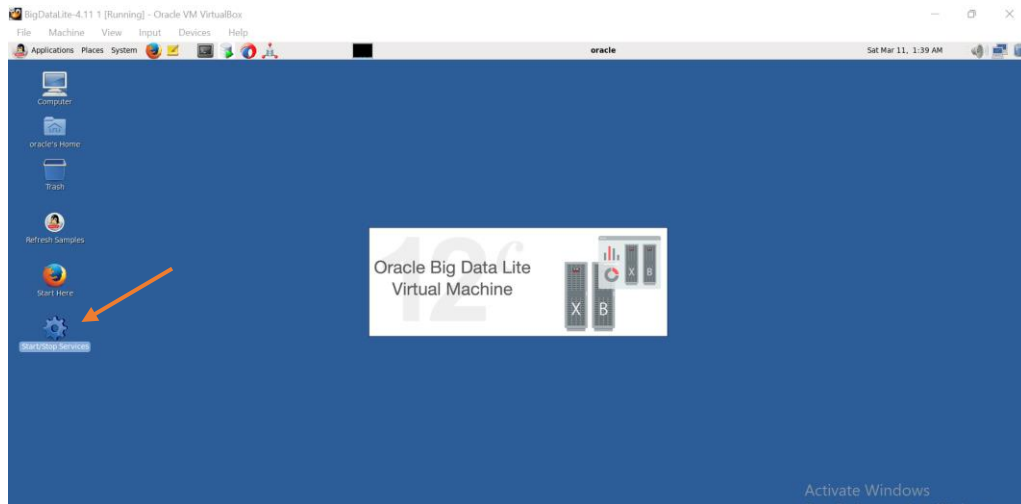
**PROGRAM STUDI SAINS DATA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”**  
**JAWA TIMUR**

**2023**

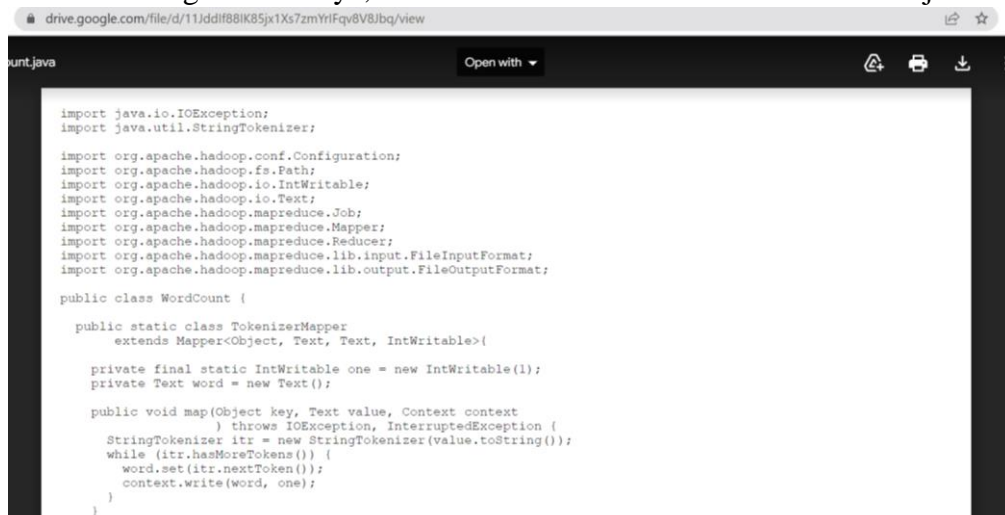
## Latihan Percobaan

Langkah-langkah:

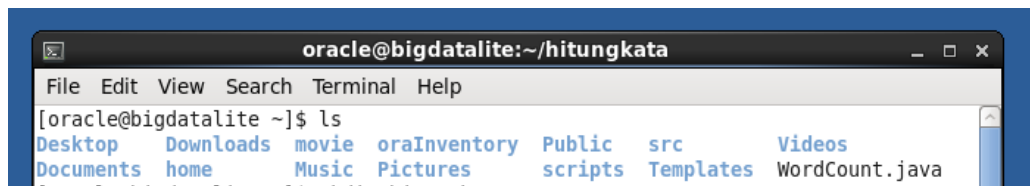
1. Install Oracle Big Data Lite
2. Untuk menggunakannya buka pada Oracle VM VirtualBox dan klik Start pada BigDataLite dan masukkan passwordnya
3. Setelah tampilannya seperti pada gambar dibawah ini, maka klik servis untuk mengaktifkan Zookeeper, HDFS, Hive, dan YARN.



4. Sebelum mengeksekusi nya, download terlebih dahulu file WordCount.java



5. Setelah itu, cek pada terminal apakah file WordCount.java sudah tersimpan dalam oracle's home



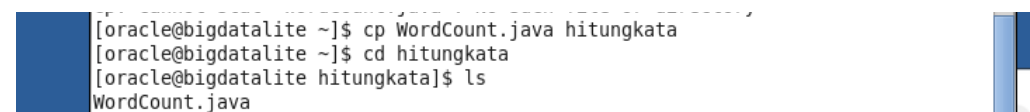
```
oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
[oracle@bigdatalite ~]$ ls
Desktop Downloads movie oraInventory Public src Videos
Documents home Music Pictures scripts Templates WordCount.java
```

6. Buat direktori baru dengan nama hitungkata menggunakan perintah mkdir



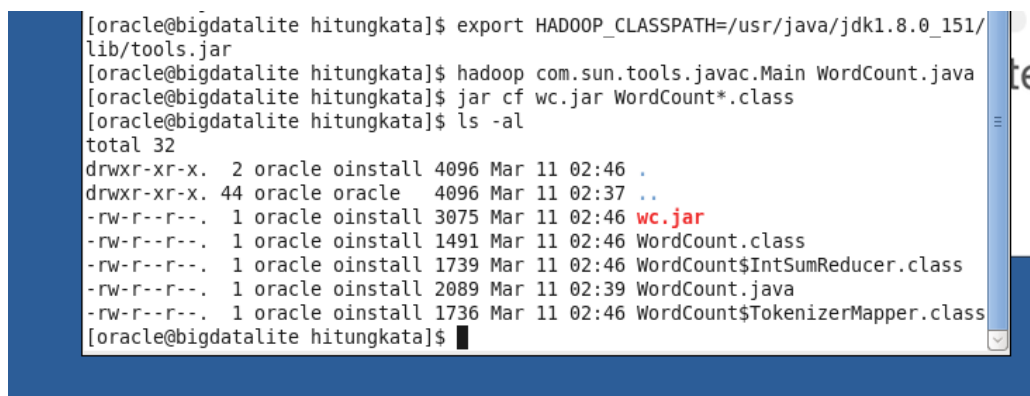
```
[oracle@bigdatalite ~]$ cd hitungkata
[oracle@bigdatalite hitungkata]$ ls
```

7. Pindahkan file WordCount.java kedalam direktori hitungkata dengan perintah cp



```
[oracle@bigdatalite ~]$ cp WordCount.java hitungkata
[oracle@bigdatalite ~]$ cd hitungkata
[oracle@bigdatalite hitungkata]$ ls
WordCount.java
```

8. Mengcompile program Java dan membuat JAR file. Pastikan kita berada di dalam direktori hitungkata



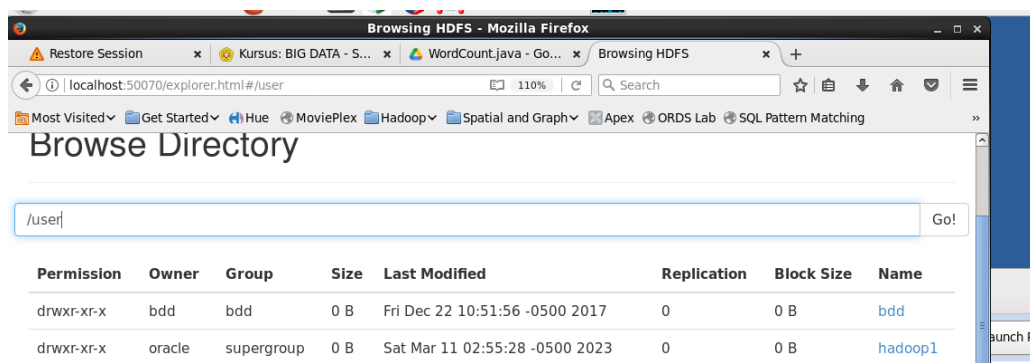
```
[oracle@bigdatalite hitungkata]$ export HADOOP_CLASSPATH=/usr/java/jdk1.8.0_151/lib/tools.jar
[oracle@bigdatalite hitungkata]$ hadoop com.sun.tools.javac.Main WordCount.java
[oracle@bigdatalite hitungkata]$ jar cf wc.jar WordCount*.class
[oracle@bigdatalite hitungkata]$ ls -al
total 32
drwxr-xr-x. 2 oracle oinstall 4096 Mar 11 02:46 .
drwxr-xr-x. 44 oracle oracle 4096 Mar 11 02:37 ..
-rw-r--r--. 1 oracle oinstall 3075 Mar 11 02:46 wc.jar
-rw-r--r--. 1 oracle oinstall 1491 Mar 11 02:46 WordCount.class
-rw-r--r--. 1 oracle oinstall 1739 Mar 11 02:46 WordCount$IntSumReducer.class
-rw-r--r--. 1 oracle oinstall 2089 Mar 11 02:39 WordCount.java
-rw-r--r--. 1 oracle oinstall 1736 Mar 11 02:46 WordCount$TokenizerMapper.class
[oracle@bigdatalite hitungkata]$
```

9. Membuat direktori sesuai user di HDFS, dan disini menggunakan nama hadoop1



```
[oracle@bigdatalite hitungkata]$ hadoop fs -mkdir hdfs:///user/hadoop1
[oracle@bigdatalite hitungkata]$
```

10. Setelah direktori telah dibuat, maka cek di browsing HDFS apa direktori sudah ada di Hadoop atau belum. Untuk membuka browsing HDFS dengan laman "localhost:50070/explorer.html#/user"



11. Buat direktori baru di Hadoop untuk menyimpan program WordCount beserta input kata yang akan di proses

```
[oracle@bigdatalite ~]$ cd hitungkata
[oracle@bigdatalite hitungkata]$ hadoop fs -mkdir hdfs:///user/hadoop1
[oracle@bigdatalite hitungkata]$ hadoop fs -ls hdfs:///user/hadoop1
[oracle@bigdatalite hitungkata]$ hadoop fs -mkdir /user/hadoop1/hitungkata
```

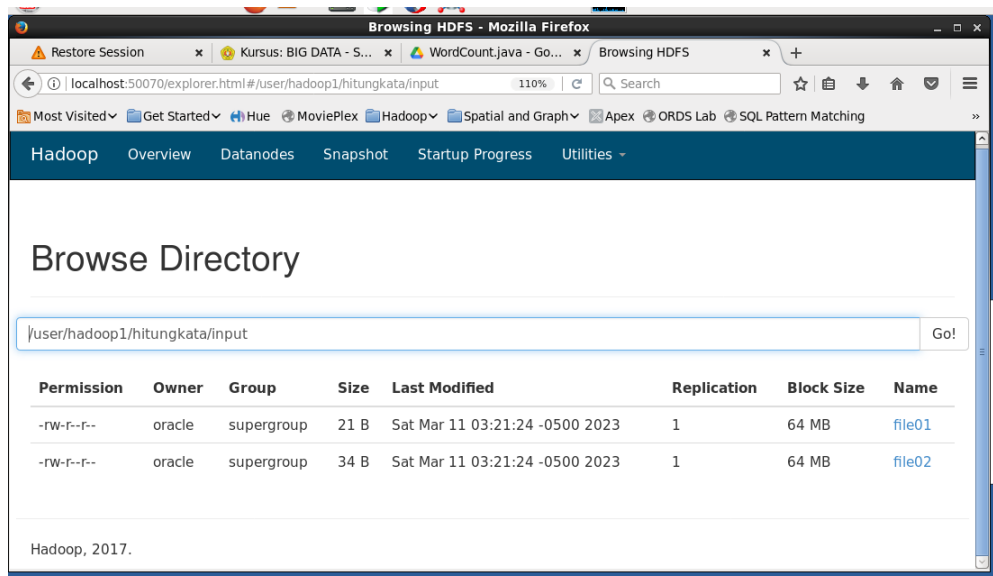
12. Buat file dengan perintah echo untuk menginputkan kalimat apapun

```
[oracle@bigdatalite hitungkata]$ hadoop fs -mkdir /user/hadoop1/hitungkata
[oracle@bigdatalite hitungkata]$ echo "hello big data" > file01
[oracle@bigdatalite hitungkata]$ echo "hello hadoop goodbye hadoop" > file02
[oracle@bigdatalite hitungkata]$ ls
file01 WordCount.class WordCount$TokenizerMapper.class
file02 WordCount$IntSumReducer.class
wc.jar WordCount.java
[oracle@bigdatalite hitungkata]$ hadoop fs -mkdir /user/hadoop1/hitungkata/input
[oracle@bigdatalite hitungkata]$ hadoop fs -copyFromLocal file0* /user/hadoop1/hitungkata/input
[oracle@bigdatalite hitungkata]$
```

13. Membuat direktori input pada Hadoop

```
[oracle@bigdatalite hitungkata]$ hadoop fs -mkdir /user/hadoop1/hitungkata/input
[oracle@bigdatalite hitungkata]$ hadoop fs -copyFromLocal file0* /user/hadoop1/hitungkata/input
[oracle@bigdatalite hitungkata]$
```

14. File tersebut berada pada lokal direktori, maka dari itu dipindahkan ke Hadoop. Kemudian cek pada browsing HDFS sudah ada atau belum



15. Untuk memastikan apakah file sudah tersimpan pada Hadoop maka dilakukan lah perintah seperti pada gambar dibawah ini

```

[oracle@bigdatalite hitungkata]$ hadoop fs -ls /user/hadoop1/hitungkata/input
Found 2 items
-rw-r--r-- 1 oracle supergroup          21 2023-03-11 03:21 /user/hadoop1/hitungkata/input/file01
-rw-r--r-- 1 oracle supergroup          34 2023-03-11 03:21 /user/hadoop1/hitungkata/input/file02
[oracle@bigdatalite hitungkata]$

```

16. Jika ingin melihat output kata yang telah buat tadi dalam terminal maka menggunakan perintah dibawah ini

```

[oracle@bigdatalite hitungkata]$ hadoop fs -cat /user/hadoop1/hitungkata/input/file01
"hello big data"
[oracle@bigdatalite hitungkata]$ hadoop fs -cat /user/hadoop1/hitungkata/input/file02
"hello hadoop goodbye hadoop"
[oracle@bigdatalite hitungkata]$

```

17. Setelah itu dilakukan JAR, dalam direktori hitungkata

```

[oracle@bigdatalite hitungkata]$ hadoop jar wc.jar WordCount /user/hadoop1/hitungkata/input /user/hadoop1/hitungkata/output
23/03/11 03:26:36 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/11 03:26:37 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/03/11 03:26:37 INFO input.FileInputFormat: Total input paths to process : 2
23/03/11 03:26:37 INFO mapreduce.JobSubmitter: number of splits:2
23/03/11 03:26:37 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678516175607_0001
23/03/11 03:26:37 INFO impl.YarnClientImpl: Submitted application application_1678516175607_0001
23/03/11 03:26:38 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_1678516175607_0001/
23/03/11 03:26:44 INFO mapreduce.Job: Running job: job_1678516175607_0001
23/03/11 03:26:44 INFO mapreduce.Job: Job job_1678516175607_0001 running in uber mode : false
23/03/11 03:26:44 INFO mapreduce.Job:  map 0% reduce 0%
23/03/11 03:26:50 INFO mapreduce.Job:  map 50% reduce 0%
23/03/11 03:26:51 INFO mapreduce.Job:  map 100% reduce 0%
23/03/11 03:26:57 INFO mapreduce.Job:  map 100% reduce 100%
23/03/11 03:26:57 INFO mapreduce.Job: Job job_1678516175607_0001 completed successfully

Reduce shuffle bytes=109
Reduce input records=7
Reduce output records=6
Spilled Records=14
Shuffled Maps=2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=170
CPU time spent (ms)=1570
Physical memory (bytes) snapshot=704057344
Virtual memory (bytes) snapshot=6287921152
Total committed heap usage (bytes)=488112128

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
Bytes Read=55
File Output Format Counters
Bytes Written=58
[oracle@bigdatalite hitungkata]$

```

18. Untuk menampilkan hasil perhitungan kata nya, dapat digunakan perintah seperti dibawah ini

```

[oracle@bigdatalite hitungkata]$ hadoop fs -cat /user/hadoop1/hitungkata/output/part*
big 1
data" 1
goodbye 1
hadoop 1
hadoop" 1
"hello 2
[oracle@bigdatalite hitungkata]$

```

19. Untuk memastikan kesuksesan program nya, dapat dilihat pada browsing HDFS nya seperti berikut

Browsing HDFS - Mozilla Firefox

localhost:50070/explorer.html#/user/hadoop1/hitungkata/output

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

### Browse Directory

/user/hadoop1/hitungkata/output Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	oracle	supergroup	0 B	Sat Mar 11 03:26:56 -0500 2023	1	64 MB	_SUCCESS
-rw-r--r--	oracle	supergroup	58 B	Sat Mar 11 03:26:56 -0500 2023	1	64 MB	part-r-00000

Hadoop, 2017.

## Tugas RTM 3

### Pada Hadoop

Menerapkan program WordCount pada 2 berita dengan 2 sumber yang berbeda dan 1 topik berita

Langkah-langkah:

1. Mencari 2 website berita. Dan disini saya mengambil berita yang baru-baru ini terjadi yaitu Erupsi Gunung Merapi di Jawa Tengah.

Link yang pertama : <https://www.detik.com/jatim/berita/d-6613351/gunung-merapi-erupsi-langit-semula-cerah-berubah-gelap-potensi-bahaya-7-km>

Link yang kedua : <https://beritadiy.pikiran-rakyat.com/citizen/pr-706406548/daftar-wilayah-yang-terkena-hujan-abu-letusan-erupsi-gunung-merapi-hari-ini-maret-2023-apa-penyebab-meletus>

2. Melakukan scrapping pada Jupyterlab Anaconda dengan kode script dibawah ini

```
from newspaper import Article

url = 'https://www.detik.com/jatim/berita/d-6613351/gunung-merapi-erupsi-langit-semula-cerah-berubah-gelap-potensi-bahaya-7-km'

article = Article(url)

article.download()

article.parse()

with open('Merapi.txt', 'w', encoding='utf-8') as f:

    f.write(article.text)
```

3. Setelah syntax berjalan dengan baik maka akan muncul isi berita dari link yang diambil kedalam sebuah file dengan format .txt

```
from newspaper import Article

url = 'https://beritadiy.pikiran-
rakyat.com/citizen/pr-706406548/daftar-wilayah-
yang-terkena-hujan-abu-letusan-erupsi-gunung-
merapi-hari-ini-maret-2023-apa-penyebab-meletus'

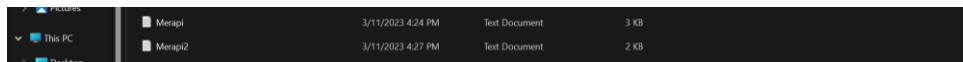
article = Article(url)

article.download()

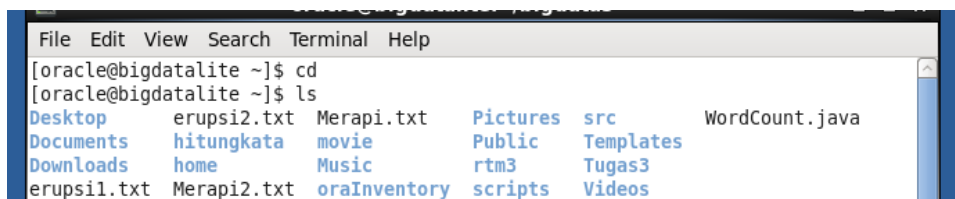
article.parse()

with open('Merapi2.txt', 'w', encoding='utf-8') as
f:

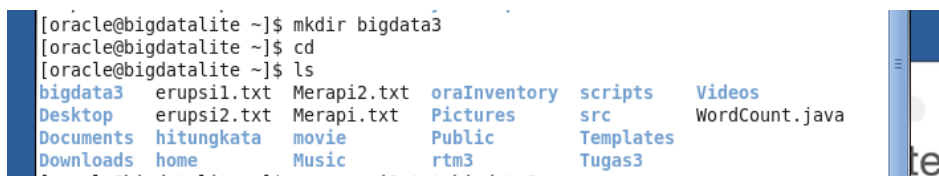
    f.write(article.text)
```



4. Setelah file .txt telah dibuat maka dipindah ke oracle's home dan untuk mengetahui file tersebut sudah ada atau belum menggunakan perintah cd dan ls untuk mengetahui isi direktori nya



5. Setelah itu dibuatlah sebuah direktori baru yang saya namai bigdata3



6. Pindahkan file erupsi1.txt, erupsi2.txt, dan WordCount.java kedalam direktory bigdata3

```

Downloads home Music rtm3 Tugas3
[oracle@bigdatalite ~]$ cp erupsi1.txt bigdata3
[oracle@bigdatalite ~]$ cp erupsi2.txt bigdata3
[oracle@bigdatalite ~]$ cp WordCount.java bigdata3
[oracle@bigdatalite ~]$ cd bigdata3
[oracle@bigdatalite bigdata3]$ ls
erupsi1.txt erupsi2.txt WordCount.java

```

7. Mengcompile program Java dan membuat JAR file. Pastikan kita berada di dalam direktori bigdata3

```

erupsi1.txt erupsi2.txt WordCount.java
[oracle@bigdatalite bigdata3]$ export HADOOP_CLASSPATH=/usr/java/jdk1.8.0_151/lib/tools.jar
[oracle@bigdatalite bigdata3]$ hadoop com.sun.tools.javac.Main WordCount.java
[oracle@bigdatalite bigdata3]$ jar cf wc.jar WordCount*.class
[oracle@bigdatalite bigdata3]$ ls -al
total 40
drwxr-xr-x. 2 oracle oinstall 4096 Mar 11 14:11 .
drwxr-xr-x. 47 oracle oracle 4096 Mar 11 14:08 ..
-rw-r--r--. 1 oracle oinstall 2380 Mar 11 14:09 erupsi1.txt
-rw-r--r--. 1 oracle oinstall 1134 Mar 11 14:09 erupsi2.txt
-rw-r--r--. 1 oracle oinstall 3075 Mar 11 14:11 wc.jar
-rw-r--r--. 1 oracle oinstall 1491 Mar 11 14:11 WordCount.class
-rw-r--r--. 1 oracle oinstall 1739 Mar 11 14:11 WordCount$IntSumReducer.class
-rw-r--r--. 1 oracle oinstall 2089 Mar 11 14:10 WordCount.java
-rw-r--r--. 1 oracle oinstall 1736 Mar 11 14:11 WordCount$TokenizerMapper.class
[oracle@bigdatalite bigdata3]$ hadoop fs -mkdir hdfs:///user/hadoop6
[oracle@bigdatalite bigdata3]$

```

8. Membuat direktori sesuai user di HDFS, dan disini menggunakan nama hadoop6

```

[oracle@bigdatalite bigdata3]$ hadoop fs -mkdir hdfs:///user/hadoop6
[oracle@bigdatalite bigdata3]$

```

9. Kemudian cek pada browsing HDFS nya

drwxr-xr-x	oracle	supergroup	0 B	Sat Mar 11 14:12:48 -0500 2023	0	0 B	<a href="#">hadoop6</a>
drwxrwxrwt	mapred	hadoop	0 B	Thu Dec 21 17:57:04 -0500 2017	0	0 B	<a href="#">history</a>
drwxrwxrwx	hive	hive	0 B	Thu Dec 21 17:31:17 -0500 2017	0	0 B	<a href="#">hive</a>

10. Buat direktori baru di Hadoop untuk menyimpan program WordCount beserta input kata yang akan di proses

```

[oracle@bigdatalite bigdata3]$ hadoop fs -mkdir hdfs:///user/hadoop6
[oracle@bigdatalite bigdata3]$ hadoop fs -ls hdfs:///user/hadoop6
[oracle@bigdatalite bigdata3]$ hadoop fs -mkdir /user/hadoop6/tugas3

```

11. Buka file erupsi1.txt dan erupsi2.txt dengan perintah nano lalu save dan keluar. Dan b egitulah isi dari berita erupsi gunung merapi di jawa tengah

```

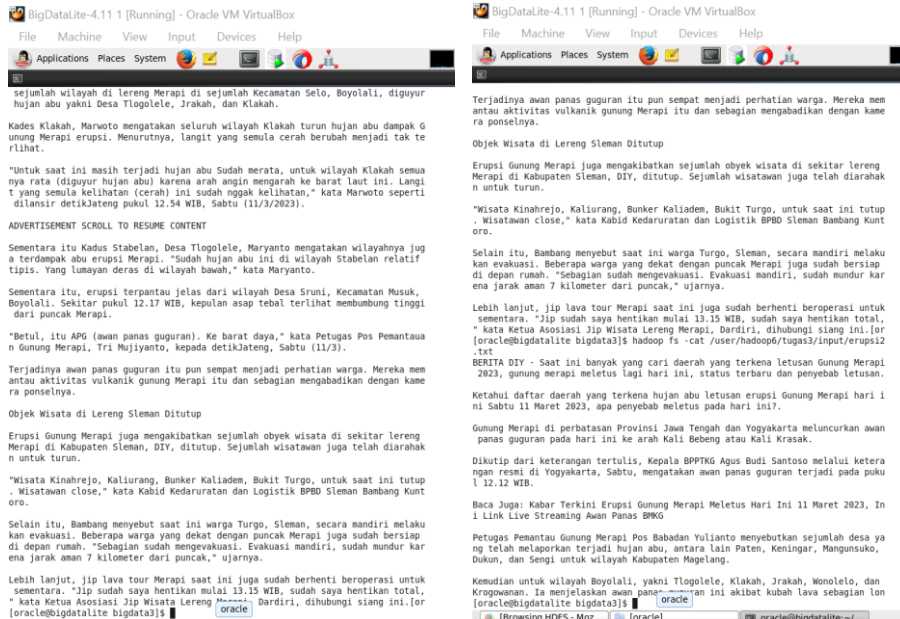
[oracle@bigdatalite bigdata3]$ nano erupsi1.txt
[oracle@bigdatalite bigdata3]$ nano erupsi2.txt
[oracle@bigdatalite bigdata3]$ nano erupsi1.txt

```

Isi berita :







## 15. Setelah itu dilakukan JAR, dalam direktori bigdata3

```

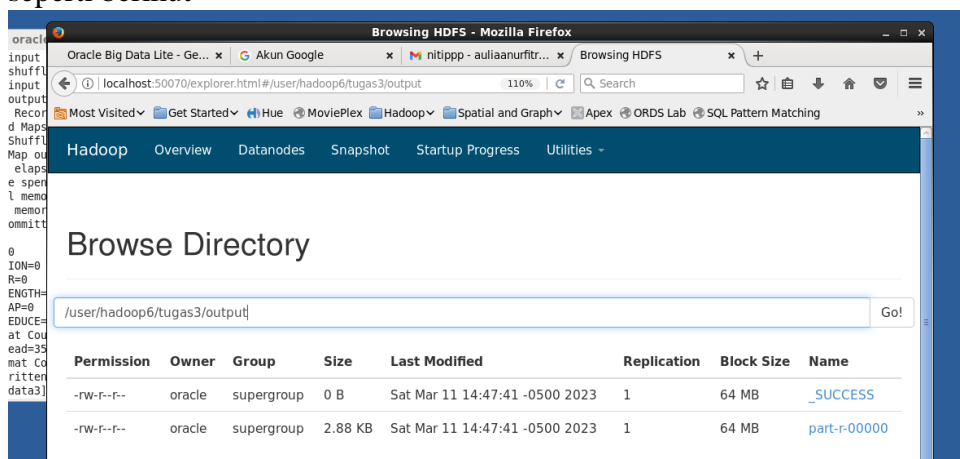
oracle@bigdatalite:~/bigdata3
Krogowanan. Ia menjelaskan awan panas guguran ini akibat kubah lava sebagian lon
[oracle@bigdatalite bigdata3]$ hadoop jar wc.jar WordCount /user/hadoop6/tugas3/
input /user/hadoop6/tugas3/output
23/03/11 14:47:19 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0
:8032
23/03/11 14:47:20 WARN mapreduce.JobResourceUploader: Hadoop command-line option
parsing not performed. Implement the Tool interface and execute your applicatio
n with ToolRunner to remedy this.
23/03/11 14:47:21 INFO input.FileInputFormat: Total input paths to process : 2
23/03/11 14:47:21 INFO mapreduce.JobSubmitter: number of splits:2
23/03/11 14:47:22 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_16
78560877230_0001
23/03/11 14:47:22 INFO impl.YarnClientImpl: Submitted application application_16
78560877230_0001
23/03/11 14:47:22 INFO mapreduce.Job: The url to track the job: http://bigdatali
te.localdomain:8088/proxy/application_1678560877230_0001/
23/03/11 14:47:22 INFO mapreduce.Job: Running job: job_1678560877230_0001
23/03/11 14:47:30 INFO mapreduce.Job: Job job_1678560877230_0001 running in uber
mode : false
23/03/11 14:47:30 INFO mapreduce.Job: map 0% reduce 0%
23/03/11 14:47:36 INFO mapreduce.Job: map 50% reduce 0%
23/03/11 14:47:37 INFO mapreduce.Job: map 100% reduce 0%
23/03/11 14:47:42 INFO mapreduce.Job: map 100% reduce 100%
23/03/11 14:47:42 INFO mapreduce.Job: Job job_1678560877230_0001 completed succe
ssfully

```

```
oracle@bigdatalite:~/bigdata3
Reduce input groups=311
Reduce shuffle bytes=4658
Reduce input records=347
Reduce output records=311
Spilled Records=694
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=205
CPU time spent (ms)=1700
Physical memory (bytes) snapshot=741949440
Virtual memory (bytes) snapshot=6282821632
Total committed heap usage (bytes)=498073600

Shuffle Errors
BAD_ID=0
CONNECTION=0
IO_ERROR=0
WRONG_LENGTH=0
WRONG_MAP=0
WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=3514
File Output Format Counters
  Bytes Written=2946
[oracle@bigdatalite bigdata3]$
```

16. Untuk memastikan kesuksesan program nya, dapat dilihat pada browsing HDFS nya seperti berikut



17. Untuk menampilkan hasil perhitungan kata nya, dapat digunakan perintah seperti dibawah ini

```
[oracle@bigdatalite bigdata3]$ hadoop fs -cat /usr
"Betul, 1
"Jip 1
"Sebagian 1
"Sudah 1
"Untuk 1
"Wisata 1
(11/3). 1
(11/3/2023). 1
(awan 1
(cerah) 1
(diguyur 1
- 1
11 2
12.12 1
12.17 1
12.54 1
13.15 1
2023, 3
7 1
ADVERTISEMENT 1
APG 1
Agus 1
Asosiasi 1
Awan 1
BERITA 1
BMKG 1
BPBD 1
BPPTKG 1
Babadan 1
Baca 1
Bambang 2
Bebeng 1
Beberapa 1
Boyolali, 2
Boyolali. 1
Budi 1
Bukit 1
Bunker 1
CONTENT 1
DIY 1
DIY, 1
Dardiri, 1
Desa 3
Dikutip 1
Ditutup 1
Dukun, 1
Erupsi 2
```

oracle

```
guguran). 1
gunung 2
hari 4
hentikan 2
hujan 7
ini 10
ini, 1
ini. 2
ini?. 1
itu 4
itu, 2
jarak 1
jelas 1
jip 1
juga 5
kamera 1
karena 2
kata 5
ke 2
kelihatan 1
kelihatan," 1
kepada 1
kepulau 1
keterangan 2
kilometer 1
kubah 1
lagi 1
lain 1
langit 1
lanjut, 1
laut 1
lava 2
lereng 2
letusan 2
letusan. 1
longsor. 1
lumayan 1
mandiri 1
mandiri, 1
masih 1
melakukan 1
melalui 1
melaporkan 1
meletus 2
meluncurkan 1
memantau 1
membuang 1
mengabadikan 1
```

```
mengevakuasi. 1
mengguyur 1
menjadi 2
menjelaskan 1
menyebut 1
menyebutkan 1
merapi 1
merata, 1
mulai 1
mundur 1
nggak 1
obyek 1
pada 3
panas 5
penyebab 2
perbatasan 2
perhatian 1
ponselnya. 1
pukul 3
pun 1
puncak 2
puncak," 1
rata 1
relatif 1
resmi 1
rumah. 1
saat 4
saya 2
```

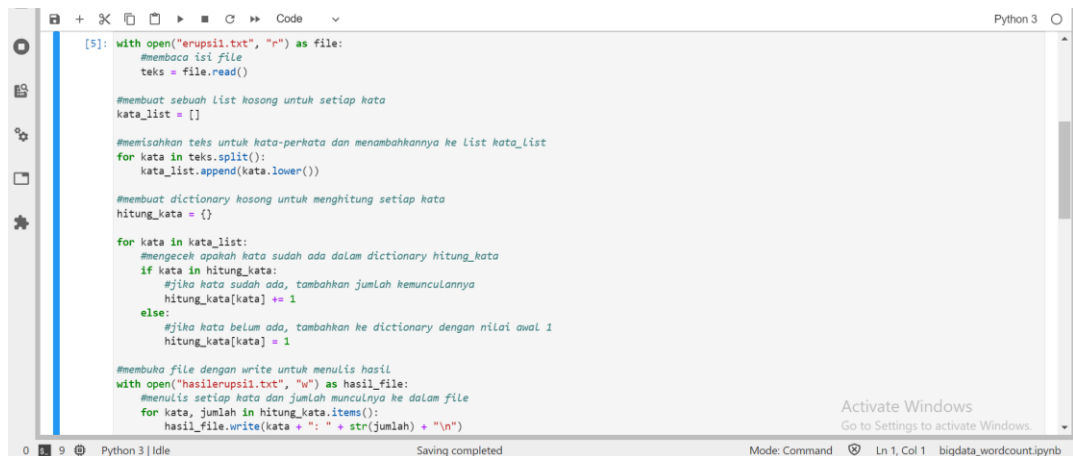
```
sebagian 2
secara 1
sejumlah 4
sekitar 1
seluruh 1
sementara. 1
sempat 2
semuanya 1
semula 2
seperti 1
siang 1
status 1
sudah 7
tak 1
tebal 1
telah 2
terbaru 1
terdampak 1
terjadi 3
terkena 2
```

## Pada Python

Syntax dibawah adalah program Python untuk menghitung frekuensi kemunculan setiap kata dalam sebuah file teks dan menyimpan hasilnya dalam sebuah file teks baru.

Dalam keseluruhan, program ini akan membuka file teks yang telah disiapkan sebelumnya, menghitung jumlah kemunculan setiap kata dalam file tersebut, dan menyimpan hasilnya dalam file teks baru dengan nama "hasilerupsi1.txt" dan "hasilerupsi2.txt"

### 1. Teks berita pertama



```
[5]: with open("erupsi1.txt", "r") as file:
    #membaca isi file
    teks = file.read()

    #membuat sebuah list kosong untuk setiap kata
    kata_list = []

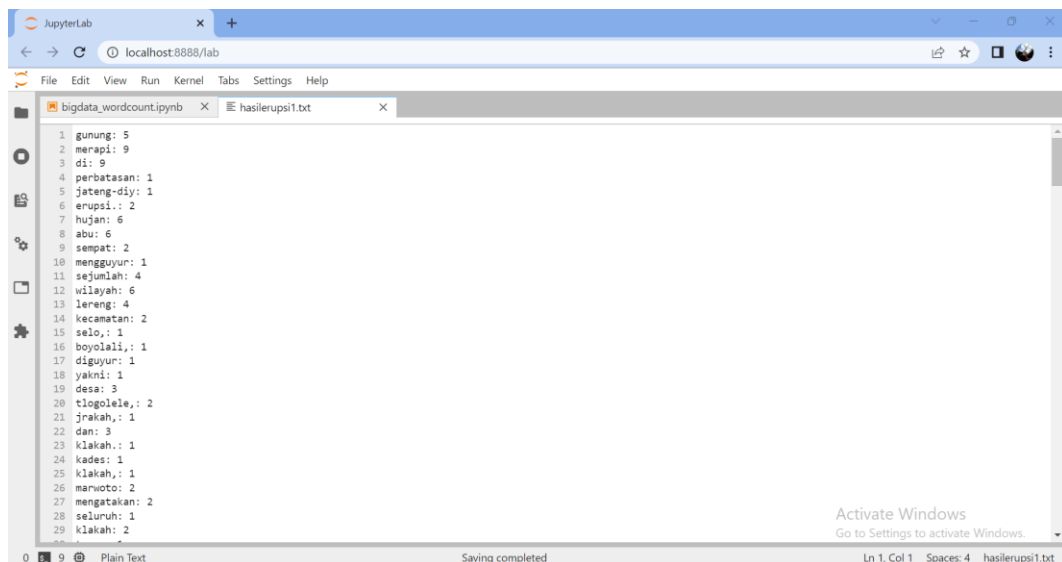
    #memisahkan teks untuk kata-perkata dan menambahkannya ke list kata_list
    for kata in teks.split():
        kata_list.append(kata.lower())

    #membuat dictionary kosong untuk menghitung setiap kata
    hitung_kata = {}

    for kata in kata_list:
        #mengecek apakah kata sudah ada dalam dictionary hitung_kata
        if kata in hitung_kata:
            #jika kata sudah ada, tambahkan jumlah kemunculannya
            hitung_kata[kata] += 1
        else:
            #jika kata belum ada, tambahkan ke dictionary dengan nilai awal 1
            hitung_kata[kata] = 1

    #membuka file dengan write untuk menulis hasil
    with open("hasilerupsi1.txt", "w") as hasil_file:
        #menulis setiap kata dan jumlah munculnya ke dalam file
        for kata, jumlah in hitung_kata.items():
            hasil_file.write(kata + ": " + str(jumlah) + "\n")
```

## Output



```
1 gunung: 5
2 merapi: 9
3 di: 9
4 perbatasan: 1
5 jateng-diy: 1
6 erupsi.: 2
7 hujan: 6
8 abu: 6
9 sempat: 2
10 mengguyur: 1
11 sejumlah: 4
12 wilayah: 6
13 lereng: 4
14 kecamatan: 2
15 selo,: 1
16 boyolali,: 1
17 diguyur: 1
18 yakni: 1
19 desa: 3
20 tlogolele,: 2
21 jrahah,: 1
22 dan: 3
23 klakah,: 1
24 kades: 1
25 klakah,: 1
26 marwoto: 2
27 mengatakan: 2
28 seluruh: 1
29 klakah: 2
```

## 2. Teks berita kedua

```
[6]: with open("erupsi2.txt", "r") as file:
    #membaca isi file
    teks = file.read()

    #membuat sebuah list kosong untuk setiap kata
    kata_list = []

    #memisahkan teks untuk kata-perkata dan menambahkannya ke list kata_list
    for kata in teks.split():
        kata_list.append(kata.lower())

    #membuat dictionary kosong untuk menghitung setiap kata
    hitung_kata = {}

    for kata in kata_list:
        #mengecek apakah kata sudah ada dalam dictionary hitung_kata
        if kata in hitung_kata:
            #jika kata sudah ada, tambahkan jumlah kemunculannya
            hitung_kata[kata] += 1
        else:
            #jika kata belum ada, tambahkan ke dictionary dengan nilai awal 1
            hitung_kata[kata] = 1

    #membuka file dengan write untuk menulis hasil
    with open("hasilerupsi2.txt", "w") as hasil_file:
        #menulis setiap kata dan jumlah munculnya ke dalam file
        for kata, jumlah in hitung_kata.items():
            hasil_file.write(kata + ": " + str(jumlah) + "\n")
```

## Output

```
1 berita: 1
2 diy: 1
3 -: 1
4 saat: 1
5 ini: 6
6 banyak: 1
7 yang: 4
8 cari: 1
9 daerah: 2
10 terkena: 2
11 letusan: 2
12 gunung: 6
13 merapi: 6
14 2023.: 3
15 meletus: 3
16 lagi: 1
17 hari: 5
18 ini.: 1
19 status: 1
20 terbaru: 1
21 dan: 4
22 penyebab: 2
23 letusan.: 1
24 ketahui: 1
25 daftar: 1
26 hujan: 2
27 abu: 1
28 erupsi: 2
29 sabtu: 1
```