

LAPORAN
RENCANA TUGAS MANDIRI (RTM) Ke-3
MATA KULIAH BIG DATA (A)
“MENDISTRIBUSIKAN DATA TEKS BERITA KE DALAM HDFS DAN
MEMBUAT MODEL PEMROGRAMAN MAPREDUCE UNTUK
PROGRAM WORDCOUNT”



DISUSUN OLEH:

Reza Putri Angga (22083010006)

DOSEN PENGAMPU:

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

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

2024

STUDI KASUS DAN PEMBAHASAN

A. STUDI KASUS

Melakukan proses membuat, menerapkan, dan menjelaskan tahap data *storage* (penyimpanan data) dan data *processing* (pemrosesan data) pada big data. Bertujuan untuk mengimplementasikan program WordCount berbasis MapReduce untuk menghitung frekuensi kata dari teks yang berada didalam 2 (dua) berita terkini sebagai *input* data. Namun, sebelum teks di distribusikan ke dalam HDFS.

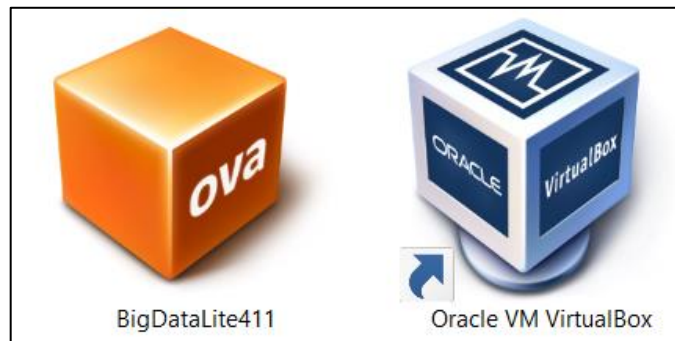
Software yang digunakan pada tugas ini adalah Oracle Big Data Lite yang didalamnya sudah ter-*install* Hadoop dan Java Environment.

B. PEMBAHASAN

Perlu diketahui bahwa untuk membahas dan menyelesaikan studi kasus tersebut, penting untuk memahami mengenai konsep atau gambaran awal terlebih dahulu. Konsep untuk melakukan perhitungan frekuensi kata ini melibatkan penggunaan *software* Oracle Big Data Lite yang didalamnya telah terintegrasi dengan Hadoop dan Java untuk menyimpan teks dari 2 berita terkini ke dalam HDFS (Hadoop Distributed File System).

Kemudian, teks tersebut akan diproses menggunakan paradigma atau alortima alur MapReduce dengan program WordCount untuk menghitung frekuensi masing-masing kata yang muncul dari 2 teks berita tersebut. Sehingga, akan dilakukan langkah-langkah, meliputi penggunaan awal Oracle Big Data Lite, 2 berita yang digunakan beserta konsep data *storage* dan data *processingnya*, serta penerapan langkah-langkah cara kerja Hadoop menggunakan program WordCount. Dengan penjelasan lebih lanjut sebagai berikut.

1. Oracle Big Data Lite

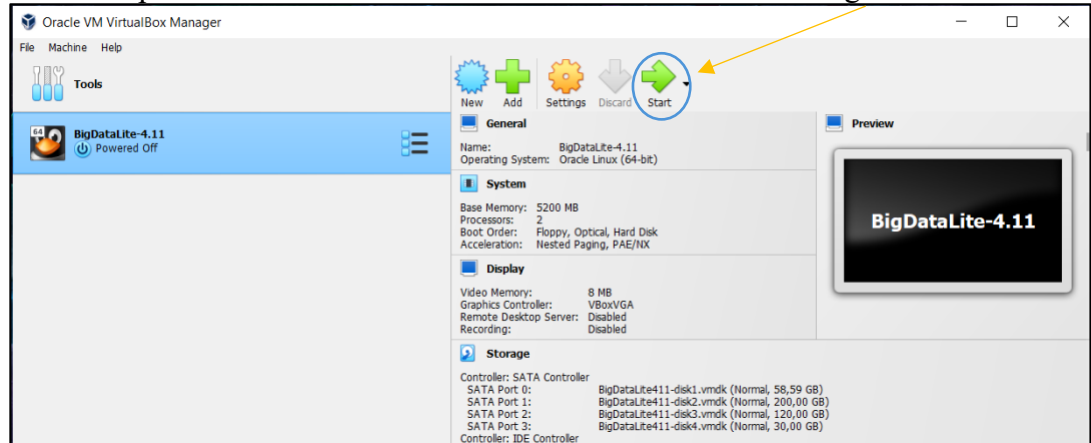


tampilan awal oracle vm virtual box dan bigdatalite411

Tahap awal dimulai dengan proses *instalasi* Oracle Big Data Lite. Terdiri dari Oracle VM VirtualBox sebagai perangkat lunak virtualisasi untuk membuat dan menjalankan sistem virtual seperti Oracle Big Data Lite, Ubuntu yang bertujuan untuk menjalankan sistem atau aplikasi tertentu dengan lingkungan virtual tanpa memengaruhi sistem operasi *laptop* atau *desktop* utama.

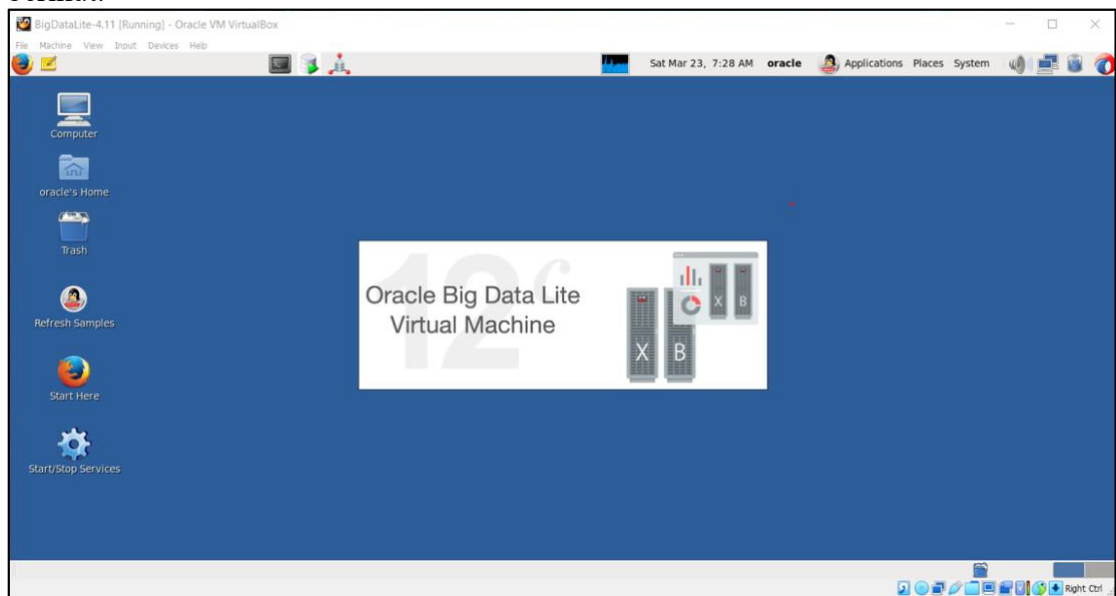
Dan, *instalasi* BigDataLite411 dengan format ova yang berisi berbagai komponen dan *tools* untuk mengembangkan dan pengujian solusi big data, seperti Hadoop, Apache Spark. Langkah selanjutnya adalah membuka Oracle VM VirtualBox dan

melakukan *add* atau *import file* BigDataLite411 kedalamnya. Setelah berhasil di *import* akan ditampilkan halaman utama dari Oracle VM VirtualBox sebagai berikut.



tampilan oracle vm virtualbox yang didalamnya ada file bigdatalite411 yang berhasil di import

Kemudian, untuk menjalankan Oracle BigDataLite dilakukan *klik* tombol “start”. Dan akan ditampilkan tampilan awal *virtual machine* dari Oracle Big Data Lite sebagai berikut.



tampilan oracle big data lite

Oracle Big Data Lite ini adalah *virtual machine*, artinya didalamnya telah berisi semua perangkat lunak dan alat yang diperlukan untuk memulai dan menjalankan program WordCount. Dengan menggunakan *virtual machine* Oracle Big Data Lite diberikan lingkungan terisolasi dan aman untuk mengembangkan, menguji, dan mempelajari teknologi big data tanpa risiko merusak sistem operasi utama.

2. 2 (Dua) Berita Yang Di Gunakan

Untuk mengimplementasikan program WordCount di dalam Oracle Big Data, digunakan dua berita terkini, yakni gempa yang terjadi di Tuban, Jawa Timur pada tanggal 22 Maret 2024. Namun, fokus utama berita yang digunakan adalah efek gempa yang terjadi di Tuban, Jawa Timur terhadap Pulau Bawean, Kabupaten Gresik, Jawa Timur.

Hal ini dikarenakan, letak gempa yang terjadi hanya berjarak sekitar 33 km dari Pulau Bawean, sehingga terjadi beberapa kerusakan akibat gempa. Dengan tampilan berita sebagai berikut.

[Gempa Bumi Dan Tsunami](#)[Layanan Publik](#)[Hubungi Kami](#)[Press Release](#)

22 Mar 2024 21:14:00 | Yana Karyana S.Kom.

ANALISIS GEOLOGI KEJADIAN GEMPA BUMI MERUSAK DI LAUT JAWA, PROVINSI JAWA TIMUR, TANGGAL 22 MARET 2024

Analisis geologi kejadian gempa bumi merusak di Laut Jawa, Provinsi Jawa Timur, sebagai berikut:

I. Informasi gempa bumi

Gempa bumi terjadi pada hari Jumat, tanggal 22 Maret 2024, pukul 11:22:45 WIB. Berdasarkan informasi dari Badan Meteorologi, Klimatologi dan Geofisika (BMKG), lokasi pusat gempa bumi terletak di Laut Jawa pada koordinat 5,74 LS dan 112,32 BT, dengan magnitudo (M6,0) pada kedalaman 10 km, berjarak sekitar 33 km barat Pulau Bawean, Kabupaten Gresik, Provinsi Jawa Timur. Menurut data The United States Geological Survey (USGS) Amerika Serikat, lokasi pusat gempa bumi terletak pada koordinat 112,311 BT dan 5,809 LS dengan magnitudo (M5,6) pada kedalaman 10 km. Berdasarkan data dari GeoForschungsZentrum (GFZ), Jerman, lokasi pusat gempa bumi berada pada koordinat 111,40 BT dan 5,85 LS, dengan magnitudo (M5,5) pada kedalaman 10 km. Kejadian gempa bumi ini diikuti oleh gempa bumi susulan.

II. Kondisi geologi dan penyebab gempa bumi

Wilayah yang terletak dekat dengan lokasi pusat gempa bumi adalah Pulau Bawean, Kabupaten Gresik, Provinsi Jawa Timur. Wilayah tersebut pada umumnya merupakan morfologi dataran hingga dataran bergelombang yang berbatasan dengan perbukitan hingga perbukitan terjal pada bagian tengahnya. Menurut data Badan Geologi (BG) pada peta geologi lembar Bawean dan Masalembu (Aziz dkk., 1993), Pulau Bawean tersusun oleh batuan berumur Tersier (terdiri – dari batupasir dan batugamping) dan dominan endapan Kuartar (terdiri – dari batuan rombakan gunung api muda dan endapan aluvial pantai). Sebagian batuan berumur Tersier tersebut telah mengalami pelapukan. Endapan Kuartar dan batuan berumur Tersier yang telah mengalami pelapukan tersebut bersifat urai, lunak, lepas, belum kompak (unconsolidated) dan memperkuat efek guncangan, sehingga rawan guncangan gempa bumi. Selain itu morfologi perbukitan yang tertutup oleh batuan berumur yang telah mengalami pelapukan akan berpotensi terjadi gerakan tanah/longsoran apabila dipicu guncangan gempa bumi kuat dan curah hujan tinggi di daerah ini.

tampilan berita satu

Berita satu diperoleh dari *website* pusat vulkanologi dan mitigasi bencana geologi dengan judul berita “Analisis Geologi Kejadian Gempa Bumi Merusak Di Laut Jawa, Provinsi Jawa Timur, Tanggal 22 Maret 2024” dengan *url* berita :

<https://vsi.esdm.go.id/kajian-kejadian/analisis-geologi-kejadian-gempa-bumi-merusak-di-laut-jawa-provinsi-jawa-timur-tanggal-22-maret-2024>

[Menu](#) [Cari](#) [Berlangganan](#) [Masuk](#)

GEMPA BUMI

Gempa Beruntun di Barat Pulau Bawean, Bangunan Rusak dan Warga Luka

Gempa beruntun di Laut Jawa sebelah barat Pulau Bawean, Jatim, menyebabkan bangunan rusak dan warga terluka.

[Audio Berita](#) 7 menit

Oleh AMBROSIUS HARTO MANUMYOYO, DAHLIA IRAWATI, DEFRI WERDIONO
22 Maret 2024 21:16 WIB - 4 menit baca

A TEKS [f](#) [t](#) [s](#) [v](#) [p](#) [b](#)



KOMPAS/BAHANA PATRISIA GUPTA

tampilan berita dua

Berita dua diperoleh dari *website* [kompas.id](https://www.kompas.id) dengan judul berita “Gempa Beruntun Di Barat Pulau Bawean, Bangunan Rusak Dan Warga Luka” dengan *url* berita :

<https://www.kompas.id/baca/nusantara/2024/03/22/gempa-beruntun-di-barat-pulau-bawean-bangunan-rusak-dan-warga-luka>

Nantinya, isi dari berita ini akan di implementasikan ke dalam konsep data *storage* dan *data processing*. Konsep data *storage* dan *data processing* pada big data dimulai dengan pengumpulan teks dari dua berita terkini. Langkah pertama, harus dilakukan penyimpanan teks sebelum teks tersebut di distribusikan ke dalam HDFS yang merupakan sistem penyimpanan terdistribusi agar dapat diakses dan diproses secara efisien.

Kemudian, dilakukan pemrosesan data dengan menghitung frekuensi kata dalam setiap teks berita menggunakan program WordCount berbasis paradigma MapReduce. MapReduce merupakan paradigma pemrograman untuk pemrosesan data paralel dalam lingkungan terdistribusi, seperti Hadoop. Terdapat beberapa tahapan MapReduce.

Pertama adalah *split* dimana berita dibagi menjadi bagian-bagian kecil. Kedua adalah *mapping* dimana setiap kata dalam *split* di *mapping* menjadi pasangan (kata, frekuensi). Ketiga adalah *shuffle* dimana pasangan-pasangan tersebut diatur ulang untuk proses reduktor yang sesuai. Dan, yang terakhir adalah *reducing* dimana frekuensi kata-kata tersebut dijumlahkan untuk menghasilkan total frekuensi kata dalam dua teks berita.

Perlu diketahui, bahwa dilakukan proses *cleaning* (pembersihan) teks berita seperti mengubah semua huruf menjadi huruf kecil *lower case*, menghapus tanda baca, dan menghapus karakter-karakter yang tidak berperan penting dan dapat menurunkan keakuratan hasil MapReduce. Untuk *cleaning* ini, digunakan bahasa pemrograman *python* dengan *function replace ()*.

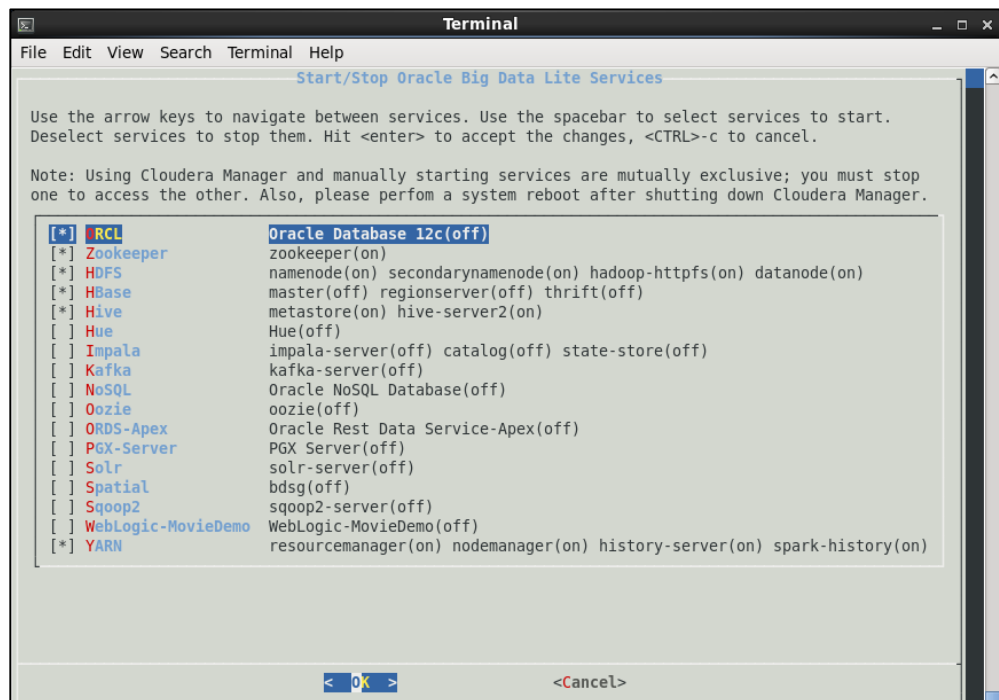
3. Penerapan Program WordCount

Untuk menerapkan program WordCount pada *file* yang terdistribusi di dalam HDFS menggunakan *software* Oracle Big Data Lite, diperlukan beberapa langkah-langkah yang akan dijalankan pada Oracle Big Data Lite, diantaranya yakni.

1. Melakukan On (Menyalakan) Pada Service

Tahap pertama untuk melakukan penerapan program WordCount dilakukan proses pengaktifan (*on*) pada *service*. Secara *default* terdapat beberapa *service* yang aktif, yakni Zookeeper, Hadoop Distributed File System (HDFS), Hive, dan Yet Another Resource Negotiator (YARN). Dimana layanan tersebut digunakan untuk infrastruktur Hadoop berfungsi secara menyeluruh untuk memproses dan menganalisis data besar.

Zookeeper digunakan untuk mengatur dan mengkoordinasikan layanan terdistribusi dalam lingkungan Hadoop, termasuk koordinasi, sinkronisasi, dan deteksi kegagalan. HDFS digunakan untuk penyimpanan terdistribusi yang digunakan oleh Hadoop untuk menyimpan data secara efisien. Hive sebagai sistem antar muka pengguna yang memungkinkan untuk melakukan analisis data menggunakan bahasa SQL.

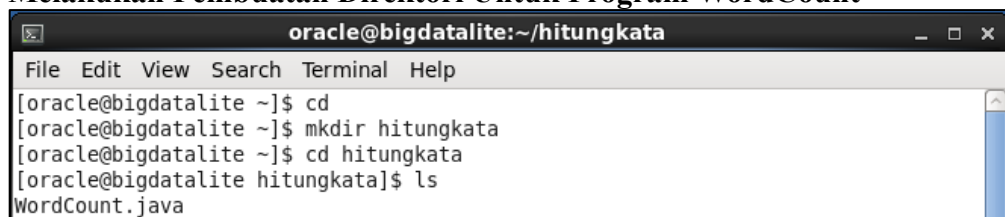


tampilan service layanan yang on atau aktif

Dan, YARN yang digunakan untuk manajer sumber daya dalam lingkungan Hadoop yang bertanggung jawab untuk penjadwalan tugas *cluster* pemrosesan data. Dalam konsep ini, HDFS yang berperan penting untuk menyimpan teks dari dua berita terkini sebelum dilakukan pemrosesan dengan program WordCount berbasis paradigma MapReduce.

Kemudian, membuka terminal untuk menjalankan beberapa perintah yang berkaitan dengan WordCount.

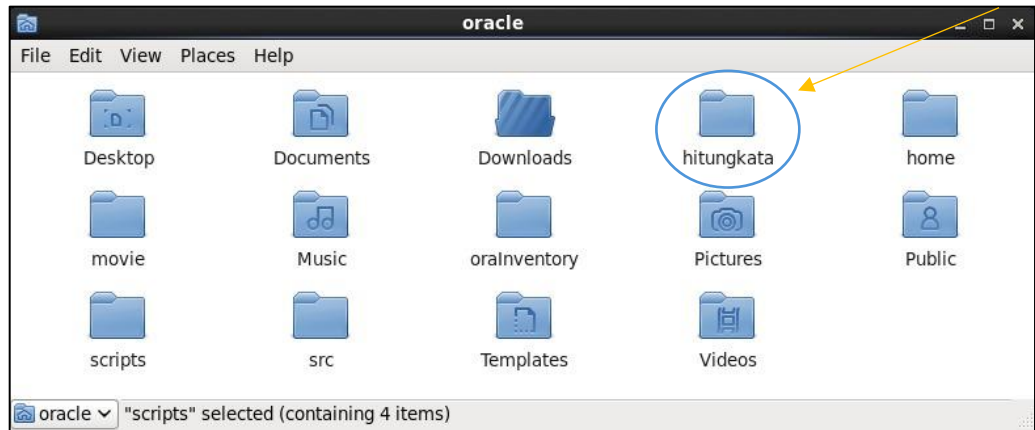
2. Melakukan Pembuatan Direktori Untuk Program WordCount



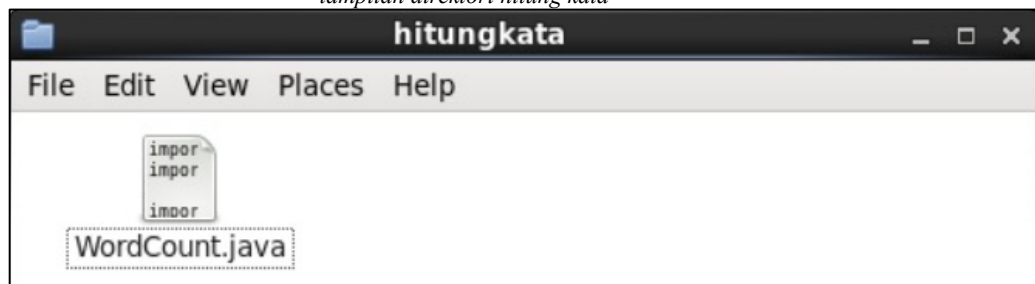
tampilan proses pembuatan direktori

Tahap kedua, dilakukan proses pembuatan direktori untuk program WordCount dengan melakukan masuk kedalam direktori utama atau *home* direktori dengan perintah *\$cd*. Kemudian, dilakukan pembuatan direktori dengan nama “hitungkata” dengan perintah *\$mkdir hitungkata*. Dan, masuk kedalam direktori hitungkata tersebut untuk memasukkan *copy file* Bernama WordCount.java.

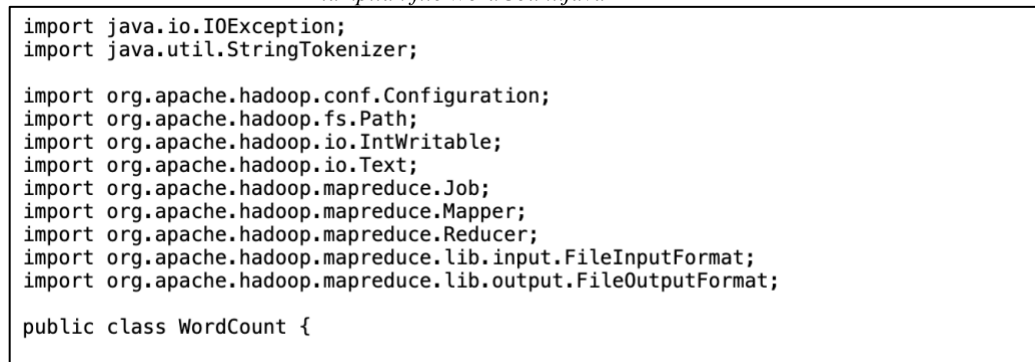
Pada proses ini, digunakan *link google drive* sebagai media untuk mengunduh *file* WordCount.java pada komputer utama ke *virtual machine*. Dilakukan pengecekan apakah *file* tersebut telah berada didalam direktori hitungkata dengan perintah *\$ls WordCount.java*.



tampilan direktori hitung kata



tampilan file WordCount.java



tampilan cuplikan isi file WordCount.java

Pada kode didalam WordCount.java berisi program WordCount dengan teks sebagai *input*, memetakan setiap kata ke nilai 1 dengan TokenizerMapper, dan menghitung total kemunculan setiap kata dengan IntSumReducer.

3. Melakukan Compile Program WordCount.Java

Tahap ketiga, dilakukan proses compile (mengkomplikasi) program java pada WordCount.java dan membuat file JAR yang merupakan formal file arsip untuk mengemas dan menyimpan satu atau lebih file kode sumber Java, file kelas (*class files*), sumber daya, dan metadata dalam kesatuan yang apat di distribusikan dengan mudah.

```
oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
[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 22 04:48 .
drwxr-xr-x. 44 oracle oracle 4096 Mar 22 04:44 ..
-rw-r--r--. 1 oracle oinstall 3075 Mar 22 04:48 wc.jar
-rw-r--r--. 1 oracle oinstall 1491 Mar 22 04:48 WordCount.class
-rw-r--r--. 1 oracle oinstall 1739 Mar 22 04:48 WordCount$IntSumReducer.class
-rw-r--r--. 1 oracle oinstall 2089 Mar 19 04:47 WordCount.java
-rw-r--r--. 1 oracle oinstall 1736 Mar 22 04:48 WordCount$TokenizerMapper.class
```

tampilan proses compile program WordCount.java

Setelah berada didalam direktori *hitungkata* dilakukan proses pengaturan variabel lingkungan `HADOOP_CLASSPATH` untuk menunjukkan lokasi *tools.jar* yang diperlukan untuk *compile* program java dengan perintah `$export HADOOP_CLASSPATH=/usr/java/jdk1.8.0_151/lib/tools.jar`. Digunakan kompilator java untuk mengubah program didalam *WordCount.java* menjadi *file* yang bisa dijalankan di Hadoop dengan perintah `$hadoop com.sun.tools.javac.Main WordCount.java`.

File yang telah dikompilasi (*compile*) digabungkan menjadi satu *file* dengan perintah `$jar cf wc.jar WordCount*.class`. Dilakukan penampilan semua *file* didalam direktori *hitungkata*, termasuk penampilan *file* jar yang telah dibuat. Dan dilakukan pengecekan apakah HDFS telah berjalan dengan baik menggunakan perintah `$sudo jps`.

```
oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
[oracle@bigdatalite hitungkata]$ sudo jps
3218 NodeManager
2770 NameNode
3107 JobHistoryServer
11443 Jps
4101 HistoryServer
3882 RunJar
2906 SecondaryNameNode
2571 QuorumPeerMain
3404 ResourceManager
2652 DataNode
3695 RunJar
3039 Bootstrap
```

tampilan proses pengecekan hadoop

4. Melakukan Pembuatan Direktori di HDFS

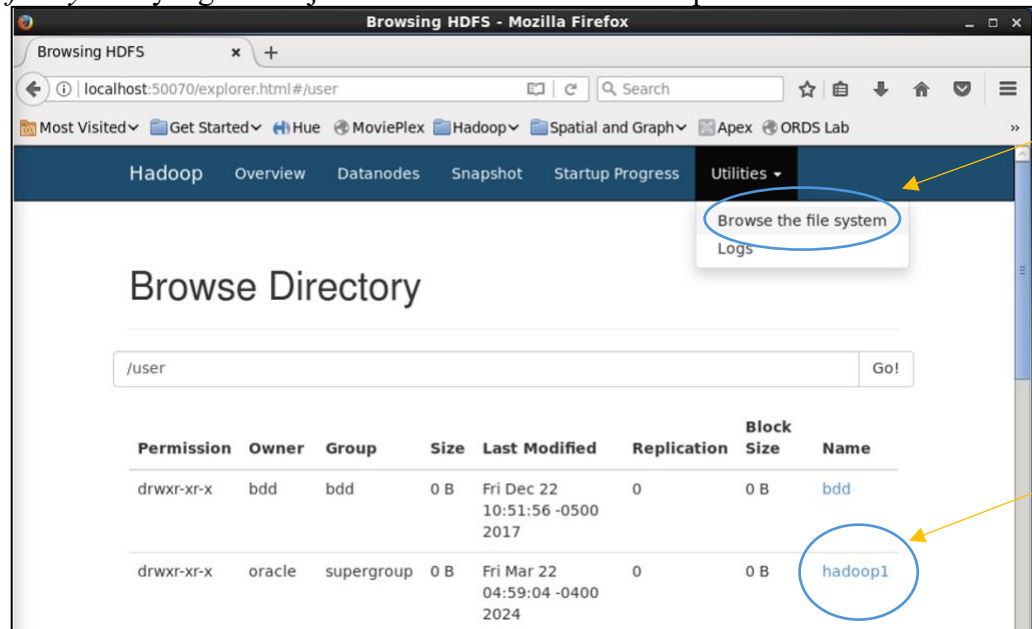
```
oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
[oracle@bigdatalite hitungkata]$ hadoop fs -mkdir hdfs:///user/hadoop1
[oracle@bigdatalite hitungkata]$
```

tampilan proses pembuatan direktori hadoop1

Tahap keempat, dilakukan pembuatan direktori di HDFS untuk menyimpan program *WordCount* dan *input* kata yang akan diproses. Dilakukan pembuatan direktori baru *hadoop1* dengan perintah `$hadoop fs -mkdir`

/user/hadoop1. Dilakukan pengecekan, apakah direktori hadoop1 telah berhasil dibuat dengan perintah `$hadoop fs -ls /user/hadoop1`.

Atau dapat dicek dengan membuka `localhost:5070 > utilities > browse the file system` yang menunjukkan bahwa direktori hadoop1 berhasil dibuat.



tampilan hasil pembuatan direktori hadoop1

Setelah berhasil membuat direktori hadoop1 di hadoop. Langkah selanjutnya adalah membuat direktori baru bernama *hitungkata* didalam direktori pengguna *hadoop1* yang akan digunakan untuk menyimpan program WordCount beserta *input* teks berita yang akan diproses dengan menggunakan perintah `$hadoop fs -mkdir hdfs:///user/hadoop1/hitungkata`.

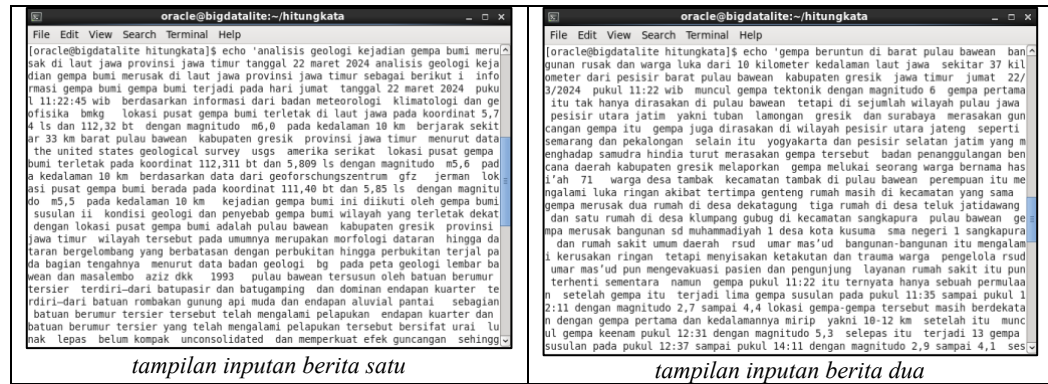


tampilan proses pembuatan direktori hitung kata didalam hadoop1

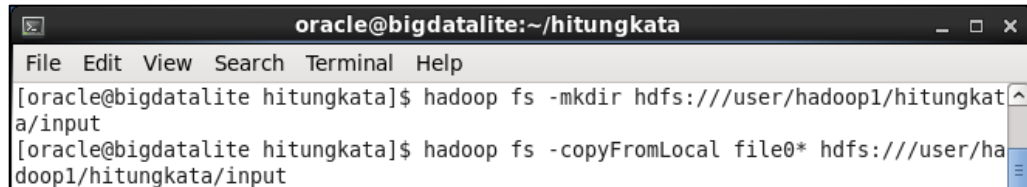
5. Melakukan Pembuatan File Input Teks Berita

Tahap kelima, dilakukan pembuatan *file input* dua teks berita terkini yang akan akan disimpan didalam *local* direktori terlebih dahulu kemudian dipindahkan ke *hitungkata* yang terdapat di HDFS *hadoop1*. Dua berita yang akan di *inputkan*, telah dilakukan *cleaning* sebelumnya, Dengan melakukan *input* teks berita yang dapat dilakukan di terminal dengan perintah `$echo isi-teks-berita > file-teks-berita` ataupun dapat dilakukan *copy paste* sama seperti program java sebelumnya.

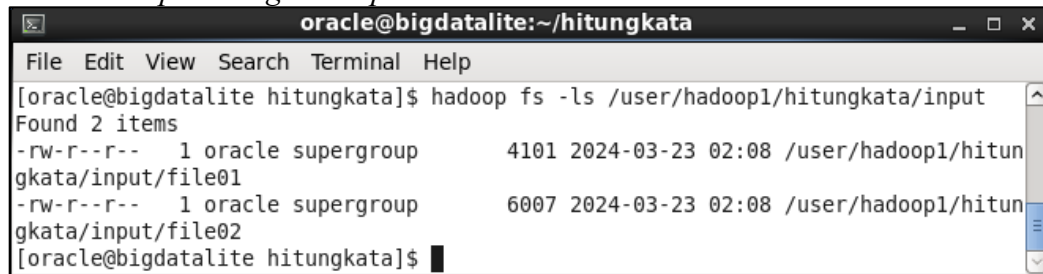
Pada praktek ini, digunakan langkah *input* teks berita satu di terminal dengan perintah `$echo 'analisis geologi kejadian gempa bumi...' > file01`, yang berarti berita tersebut disimpan dalam *file01*. Dan *input* teks berita dua di terminal dengan perintah `$echo 'gempa beruntun di barat pulau bawean...' > file02`, yang berarti berita tersebut disimpan dalam *file02*.



Setelah berhasil melakukan *inputan* kedua teks berita tersebut didalam *local* direktori, selanjutnya akan dilakukan pemindahan dua *file* teks berita dengan nama *file01* dan *file02* ke dalam input yang dibuat dengan perintah *\$hadoop fs-mkdir hdfs://user/hadoop1/hitungkata/input*. Dan dilakukan *copy file input* tersebut ke direktori *hitungkata* di *hadoop* dengan *\$hadoop fs-copyFromLocal file0* hdfs://user/hadoop1/hitungkata/input*.



Dan memastikan bahwa kedua teks berita *file01* dan *file02* telah berhasil *terinput* kedalam *hadoop1>hitungkata>input* dengan perintah *\$hadoop fs -ls /user/hadoop1/hitungkata/input*.



6. Melakukan Menjalankan JAR

Tahap keenam, dilakukan proses menjalankan file jar dari WordCount yang sebelumnya sudah dibuat file jarnya. Pastikan telah berada didalam direktori *hitungkata* di *hadoop1* lalu jalankan *file jar* dengan perintah *\$hadoop jar wc.jar WordCount /user/hadoop1/hitungkata/input /user/hadoop1/hitungkata/output*. Dimana perintah tersebut menjalankan tugas sebagai berikut, *hadoop jar wc.jar* untuk menjalankan program jar menggunakan infrastruktur hadoop.

WordCount sebagai *main class* dari program didalam file jar yang ingin dijalankan. */user/hadoop1/hitungkata/input* sebagai direktori *inputan* program Wordcount berupa *file01* dan *file02* yang berisi teks berita. Dan, */user/hadoop1/hitungkata/output* adalah direktori *output* yang akan disimpan.

```
oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
[oracle@bigdatalite hitungkata]$ hadoop jar wc.jar WordCount /user/hadoop1/hitungkata/input /user/hadoop1/hitungkata/output
24/03/23 02:14:59 INFO client.RMPProxy: Connecting to ResourceManager at /0.0.0.0:8032
24/03/23 02:15:01 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
24/03/23 02:15:03 INFO input.FileInputFormat: Total input paths to process : 2
24/03/23 02:15:03 INFO mapreduce.JobSubmitter: number of splits:2
24/03/23 02:15:04 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1711163901380_0002
24/03/23 02:15:05 INFO impl.YarnClientImpl: Submitted application application_1711163901380_0002
24/03/23 02:15:05 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_1711163901380_0002/
24/03/23 02:15:05 INFO mapreduce.Job: Running job: job_1711163901380_0002
24/03/23 02:15:25 INFO mapreduce.Job: Job job_1711163901380_0002 running in uber mode : false
24/03/23 02:15:25 INFO mapreduce.Job: map 0% reduce 0%
24/03/23 02:15:42 INFO mapreduce.Job: map 50% reduce 0%
24/03/23 02:15:44 INFO mapreduce.Job: map 100% reduce 0%
24/03/23 02:15:57 INFO mapreduce.Job: map 100% reduce 100%
24/03/23 02:15:58 INFO mapreduce.Job: Job job_1711163901380_0002 completed successfully
24/03/23 02:15:59 INFO mapreduce.Job: Counters: 49
File System Counters
FILE: Number of bytes read=8854
FILE: Number of bytes written=452929
FILE: Number of read operations=0
FILE: Number of large read operations=0
FILE: Number of write operations=0
```

Request Job
MapReduce
Success

tampilan proses request mapreduce

Program MapReduce ini digunakan untuk menghitung jumlah kata yang muncul dalam sebuah *file* teks berita. Dengan tahapan MapReduce yang telah dijelaskan pada konsep data *storage* dan data *processing*. Kemudian, untuk proses MapReduce dapat ditampilkan sebagai berikut.

```
oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
Map-Reduce Framework
Map input records=2
Map output records=1401
Map output bytes=15484
Map output materialized bytes=8860
Input split bytes=274
Combine input records=1401
Combine output records=646
Reduce input groups=561
Reduce shuffle bytes=8860
Reduce input records=646
Reduce output records=561
Spilled Records=1292
Shuffled Maps =2
Failed Shuffles=0
Merged Map outputs=2
GC time elapsed (ms)=622
CPU time spent (ms)=5470
Physical memory (bytes) snapshot=719306752
Virtual memory (bytes) snapshot=6289870848
Total committed heap usage (bytes)=500695040
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=10108
File Output Format Counters
Bytes Written=5487
```

Process
MapReduce

tampilan proses mapreduce

Pada proses MapReduce tersebut ditemukan jumlah *inputan* sebanyak 2 yang terdiri dari *file01* dan *file02*, ditampilkan hasil *output reduce* dari proses MapReduce sebesar 561.

7. Melakukan Penampilan Hasil Proses MapReduce

```

oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
Bytes Written=5487
[oracle@bigdatalite hitungkata]$ hadoop fs -cat /user/hadoop1/hitungkata/output/
part*

```

tampilan proses melihat hasil mapreduce lewat terminal

Langkah ketujuh, dilakukan penampilan hasil proses MapReduce menggunakan terminal dengan perintah `$hadoop fs -cat` untuk menampilkan isi file didalam HDFS dengan `/user/hadoop1/hitungkata/output/part*` untuk menampilkan file di dalam direktori `output` dalam `hitungkata` di HDFS. Dan ditampilkan cuplikan `output` hasil proses MapReduce didalam terminal sebagai berikut.

```

oracle@bigdatalite:~/hitungkata
File Edit View Search Terminal Help
bg 3
biasa 1
biasanya 1
bmkg 4
bpbd 5
brawijaya 1
bt 3
bumi 28
camat 1
collateral 1
cukup 2
curah 1
daerah 5
dagangan 1
dalam 2
dampak 1
dan 45
dangkal 3
dapat 2
daratan 1
dari 11

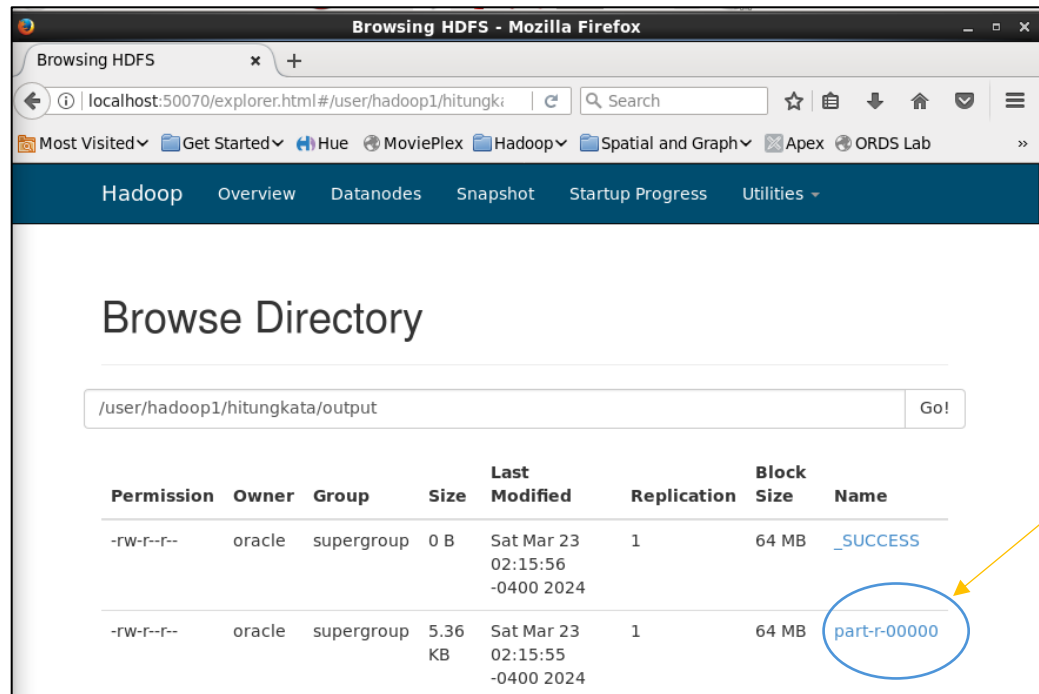
```

tampilan beberapa hasil mapreduce lewat terminal

badan 4	efek 1	trauma 2
bagi 1	endapan 3	tsunami 7
bagian 1	garis 1	tua 1
bahaya 1	gelombang 8	tuban 4
bahwa 1	gempa 64	turut 1
balai 1	gempa-gempa 1	ujar 1
bangilan 1	genteng 2	ujarnya 1
bangunan 8	geofisika 2	ulama 1
bangunan-bangunan	geoforschungszentrum	umar 2
banyaknya 1	geologi 6	umum 1
barat 5	geological 1	umumnya 1
batuan 5	gerakan 2	unair 1
batugamping 1	geser 2	unconsolidated
batupasir 1	gfz 1	united 1
bawean 16	glagahsari 1	universitas
beberapa 1	gresik 8	
belum 2	gubug 1	
bencana 5	guncangan 7	
berada 2	gunung 1	

tampilan beberapa hasil mapreduce

Ataupun, untuk melakukan pengecekan hasil *output* dapat dilakukan di *browse directory* HDFS dengan tampilan sebagai berikut.

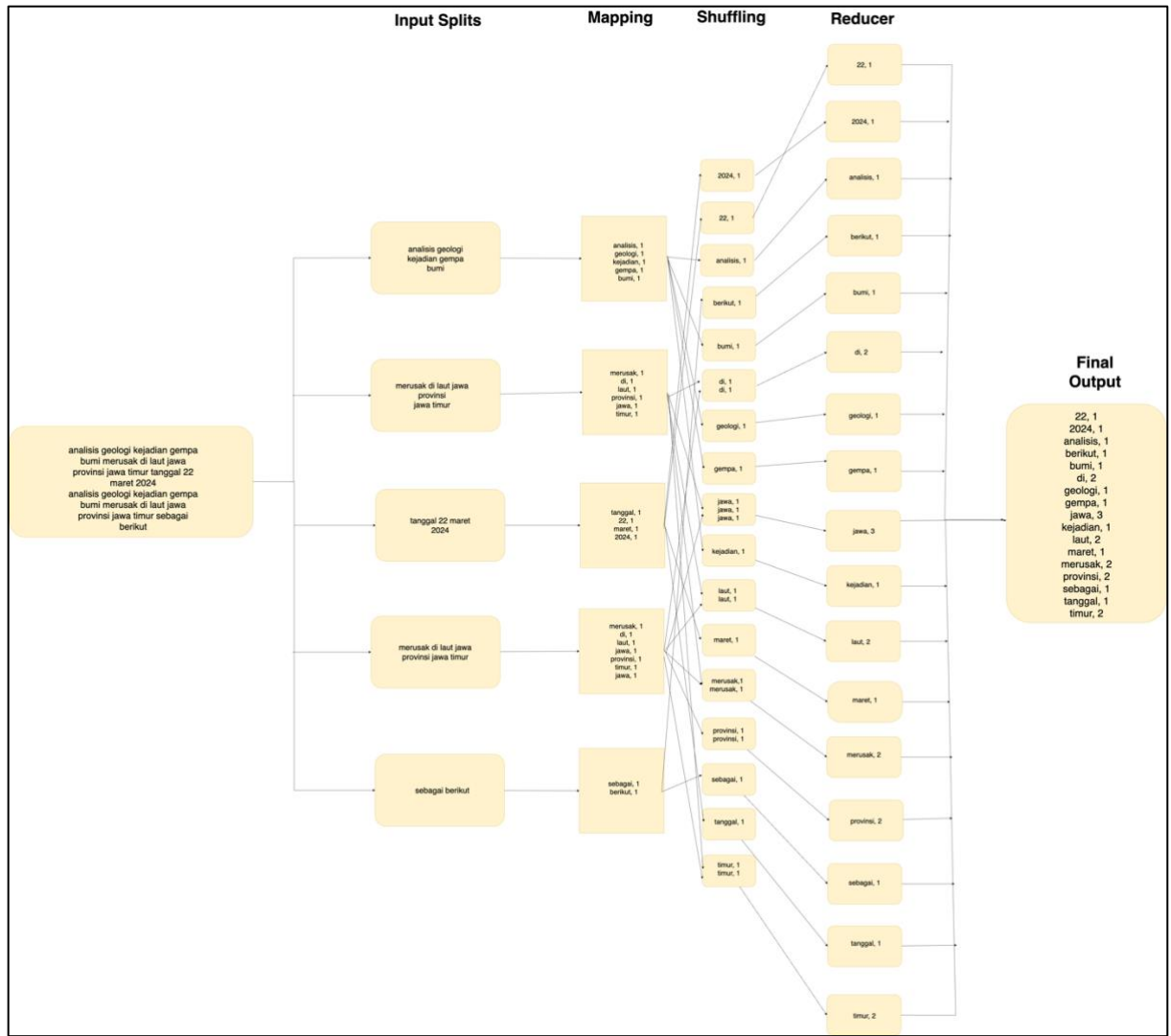


tampilan output di browse direktori HDFS

File output tersebut dapat diunduh. Kemudian, untuk memindahkan *file output* tersebut ke dalam komputer digunakan media pengiriman melalui *email*.

8. Alur Map Reduce

Untuk gambaran pembagian proses MapReduce dapat direpresentasikan sebagai berikut.



tampilan alur proses MapReduce

Kesimpulan :

Pada penugasan mendistribusikan data teks berita ke dalam HDFS dan membuat model pemrograman MapReduce untuk program WordCount ini telah berhasil dilakukan. Prosesnya diawali dengan *instalasi* Oracle Big Data Lite dengan melimpiti *import file* BigDataLite411 ke Oracle VM VirtualBox, membuat direktori, mengimpor teks berita ke HDFS, meng-*compile* program WordCount, dan menjalankannya menggunakan MapReduce. Diperoleh hasil berupa *output* frekuensi jumlah kata yang muncul didalam dua teks berita tersebut yang dihitung menggunakan MapReduce.

Keseluruhan proses ini memberikan pemahaman yang baik mengenai konsep data *storage* dan data *processing* pada big data, serta penerapan hadoop dan MapReduce dalam analisis teks berita.

Link google drive (laporan yang belum di compress, file input output, dan alur proses MapReduce) :

<https://drive.google.com/drive/folders/1rCy25-eEbOuGJb987jrUXA4KkmidCV?usp=sharing>