

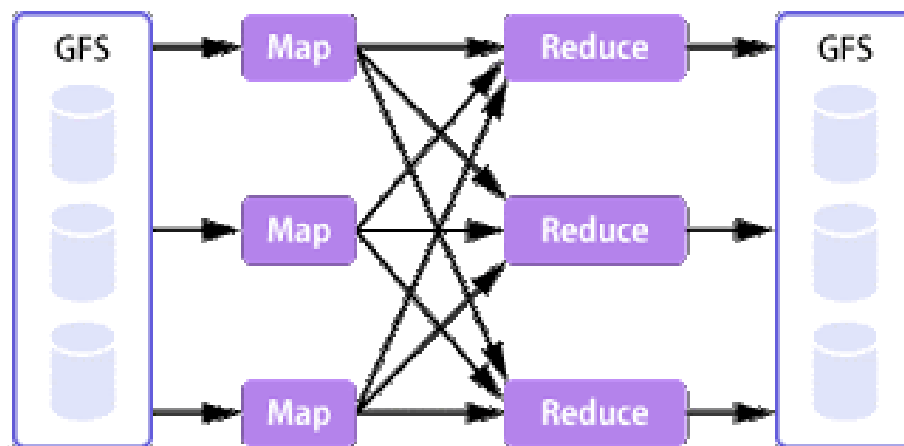
Modul 9

Implementasi Word Count pada Hadoop

RINGKASAN MATERI:

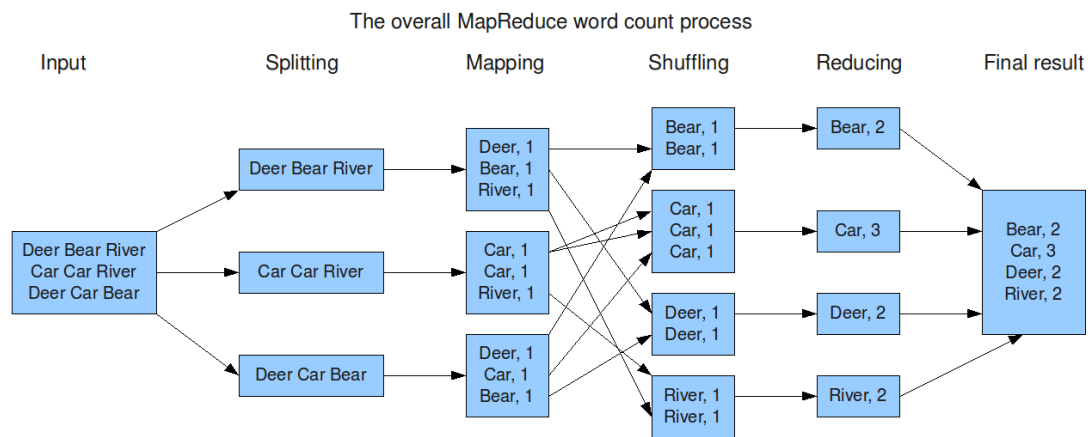
Word Count Program, merupakan salah satu program untuk menghitung jumlah kata unik dengan memanfaatkan MapReduce pada Hadoop. **MapReduce** sendiri merupakan model pemrograman rilis Google yang ditujukan untuk memproses data berukuran raksasa secara terdistribusi dan paralel dalam cluster yang terdiri atas ribuan komputer. Dalam memproses data, secara garis besar MapReduce dapat dibagi dalam dua proses yaitu proses Map dan proses Reduce.

Kedua jenis proses ini didistribusikan atau dibagi-bagikan ke setiap komputer dalam suatu cluster (kelompok komputer yang saling terhubung) dan berjalan secara paralel tanpa saling bergantung satu dengan yang lainnya. Proses Map bertugas untuk mengumpulkan informasi dari potongan-potongan data yang terdistribusi dalam tiap komputer dalam cluster. Hasilnya diserahkan kepada proses Reduce untuk diproses lebih lanjut. Hasil proses Reduce merupakan hasil akhir yang dikirim ke pengguna.



Gambar Desain Map dan Reduce

Program yang memuat kalkulasi yang akan dilakukan dalam proses Map disebut Fungsi Map, dan yang memuat kalkulasi yang akan dikerjakan oleh proses Reduce disebut Fungsi Reduce. Fungsi Map bertugas untuk membaca input dalam bentuk pasangan Key/Value, lalu menghasilkan output berupa pasangan Key/Value juga. Pasangan Key/Value hasil fungsi Map ini disebut pasangan Key/Value intermediate. Kemudian, fungsi Reduce akan membaca pasangan Key/Value intermediate hasil fungsi Map, dan menggabungkan atau mengelompokkannya berdasarkan Key tersebut. Lain katanya, tiap Value yang memiliki Key yang sama akan digabungkan dalam satu kelompok. Fungsi Reduce juga menghasilkan output berupa pasangan Key/Value.



Gambar Alur MapReduce pada WordCount

LANGKAH-LANGKAH PERSIAPAN:

(Disclaimer)

- JANGAN LUPA untuk membuat snapshot di beberapa kondisi, khususnya ketika melakukan instalasi atau hal-hal yang sekiranya dirasa penting bagi praktikan agar jika terjadi error dapat melakukan *recovery* dan *error handling* dengan mudah.
- Perhatikan PATH serta PENAMAAN masing-masing file, sesuaikan dengan yang sudah diterima.

Langkah WordCount

1. Menggunakan user **hduser**, buka terminal dan buka *file* bashrc untuk memasukkan *variable environment* baru.

```
sudo nano ~/.bashrc
```

2. Tambahkan baris berikut pada akhir, sesuaikan dengan path instalasi masing-masing yang telah dilakukan pada modul sebelumnya.

```
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/jrt-fs.jar
```



```
hduser@bigdata-VirtualBox: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /home/hduser/.bashrc Modified

export HADOOP_HOME=/home/hduser/hadoop-3.3.0
export HADOOP_INSTALL=$HADOOP_HOME
export HADOOP_MAPRED_HOME=$HADOOP_HOME
export HADOOP_COMMON_HOME=$HADOOP_HOME
export HADOOP_HDFS_HOME=$HADOOP_HOME
export YARN_HOME=$HADOOP_HOME
export HADOOP_COMMON_LIB_NATIVE_DIR=$HADOOP_HOME/lib/native
export PATH=$PATH:$HADOOP_HOME/sbin:$HADOOP_HOME/bin
export HADOOP_OPTS="-Djava.library.path=$HADOOP_HOME/lib/native"
export JAVA_HOME=/usr/lib/jvm/java-11-openjdk-amd64
export PATH=${JAVA_HOME}/bin:${PATH}
export HADOOP_CLASSPATH=${JAVA_HOME}/lib/jrt-fs.jar

Save modified buffer? (Answering "No" will DISCARD changes.)
Y Yes
N No ^C Cancel
```

Keluar dari window editing dan simpan dokumen dengan `ctrl + X` kemudian tuliskan `Y` untuk menyimpan dokumen (akan disimpan secara *overwrite*).

3. Update dengan perintah

```
source ~/.bashrc
```
4. Cek apakah *variable environment* sudah masuk dengan perintah

```
printenv JAVA_HOME
```

```
hduser@bigdata-VirtualBox:~$ printenv JAVA_HOME
/usr/lib/jvm/java-11-openjdk-amd64
hduser@bigdata-VirtualBox:~$
```

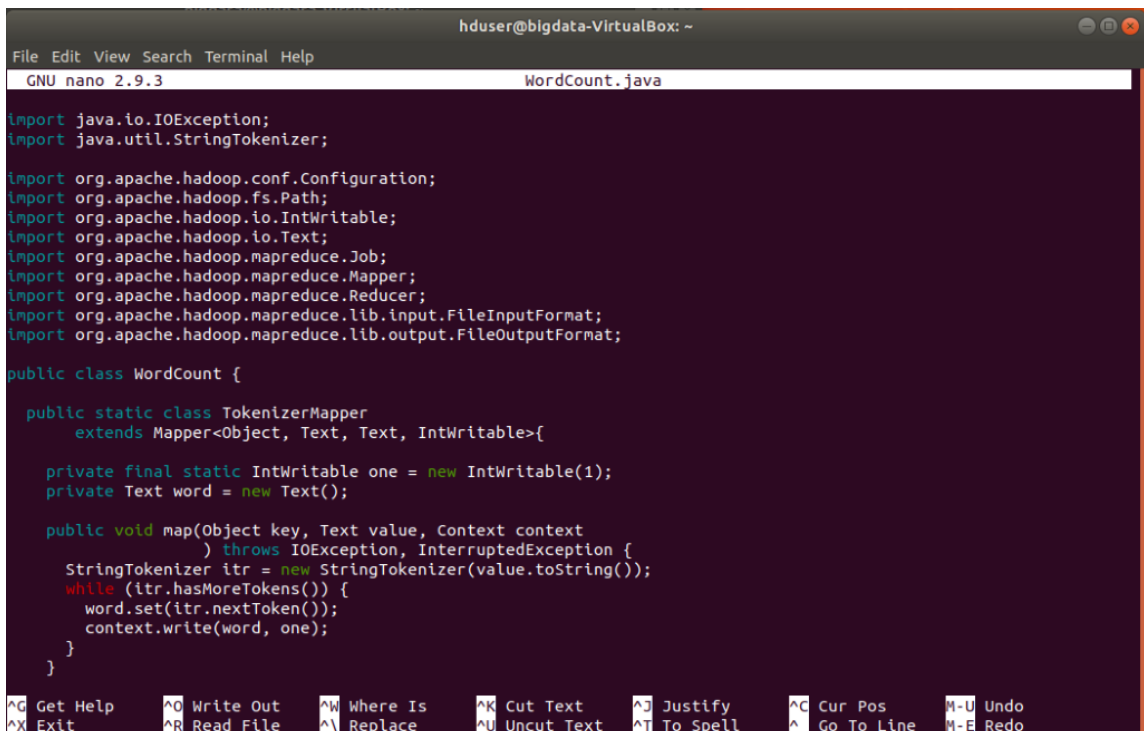
5. Membuat berkas .java baru dengan nama WordCount menggunakan perintah
`sudo nano WordCount.java`

```
hduser@bigdata-VirtualBox:~$ sudo nano WordCount.java
```

6. Lakukan pengisian dokumen WordCount.java tersebut dengan code yang diambil dari hadoop apache mapreduce pada halaman berikut:

[https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#:~:text=WordCount%20is%20a%20simple%20application,installation%20\(Single%20Node%20Setup\).](https://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html#:~:text=WordCount%20is%20a%20simple%20application,installation%20(Single%20Node%20Setup).)

Code yang diambil merupakan SELURUH code sampai bagian method Main pada bagian Source Code dalam halaman tersebut. Contohnya seperti gambar di bawah ini (baris code di gambar belum se-lengkap keseluruhan Source Code).



```
hduser@bigdata-VirtualBox: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 WordCount.java

import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

    public static class TokenizerMapper
        extends Mapper<Object, Text, Text, IntWritable>{

        private final static IntWritable one = new IntWritable(1);
        private Text word = new Text();

        public void map(Object key, Text value, Context context
            ) throws IOException, InterruptedException {
            StringTokenizer itr = new StringTokenizer(value.toString());
            while (itr.hasMoreTokens()) {
                word.set(itr.nextToken());
                context.write(word, one);
            }
        }
    }
}

^G Get Help      ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo
^X Exit          ^R Read File    ^_ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line    M-E Redo
```

7. *Compile* wordcount.java dan membuat *file* jar dengan perintah

```
hadoop com.sun.tools.javac.Main WordCount.java
```

```
hduser@bigdata-VirtualBox:~$ hadoop com.sun.tools.javac.Main WordCount.java
```

```
jar cf wc.jar WordCount*.class
```

```
hduser@bigdata-VirtualBox:~$ jar cf wc.jar WordCount*.class
```

8. Periksa apakah *file* jar sudah terbuat dengan perintah `ls`

Hasilnya dapat disesuaikan dengan gambar berikut.

```
hduser@bigdata-VirtualBox:~$ ls
examples.desktop      text.txt              'WordCount$IntSumReducer.class'
hadoop-3.3.0          tmpdata              WordCount.java
hadoop-3.3.0.tar.gz   wc.jar               'WordCount$TokenizerMapper.class'
textkopi.txt          WordCount.class
```

9. Buat folder `/user/modul7/input` dan `/user/modul7/output` pada hdfs sesuai gambar di bawah ini

```
hadoop fs -mkdir /user/modul7
```

```
hadoop fs -mkdir /user/modul7/input
```

```
hadoop fs -mkdir /user/modul7/output
```

```
hduser@bigdata-VirtualBox:~$ hadoop fs -mkdir /user/modul7
hduser@bigdata-VirtualBox:~$ hadoop fs -mkdir /user/modul7/input
hduser@bigdata-VirtualBox:~$ hadoop fs -mkdir /user/modul7/output
```

10. Periksa apakah folder sudah terbuat dengan perintah berikut.

```
hadoop fs -ls /user/modul7
```

Hasil keluaran dapat disesuaikan dengan gambar berikut.

```
hduser@bigdata-VirtualBox:~$ hadoop fs -ls /user/modul7
Found 2 items
drwxr-xr-x  - hduser supergroup      0 2020-11-14 16:03 /user/modul7/input
drwxr-xr-x  - hduser supergroup      0 2020-11-14 16:03 /user/modul7/output
```

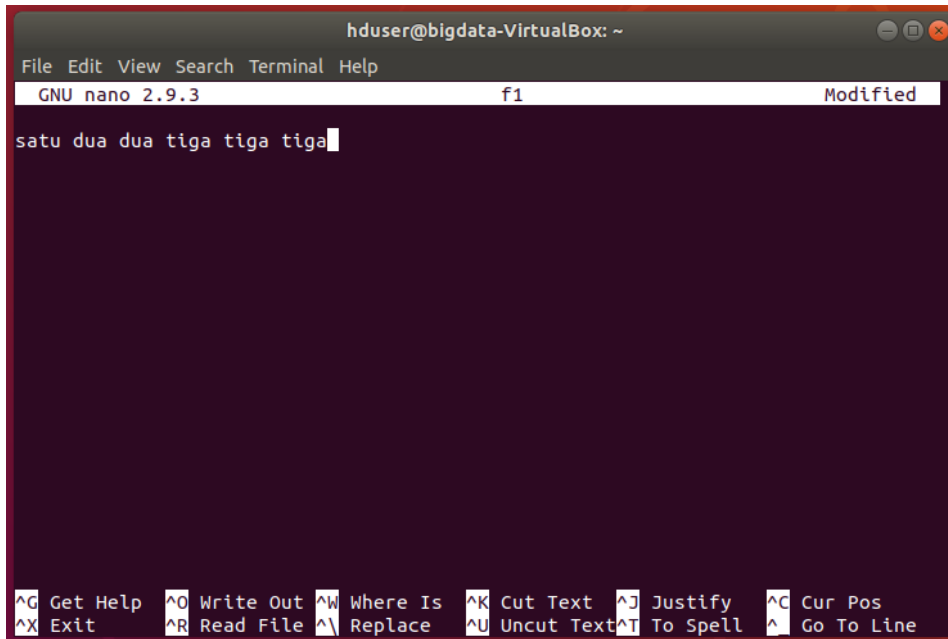
11. Membuat berkas/file baru dengan nama `f1` menggunakan perintah

```
sudo nano f1
```

```
hduser@bigdata-VirtualBox:~$ sudo nano f1
[sudo] password for hduser:
```

Dalam berkas/file f1 tersebut, isi dengan teks di bawah ini sesuai gambar yang disediakan

satu dua dua tiga tiga tiga



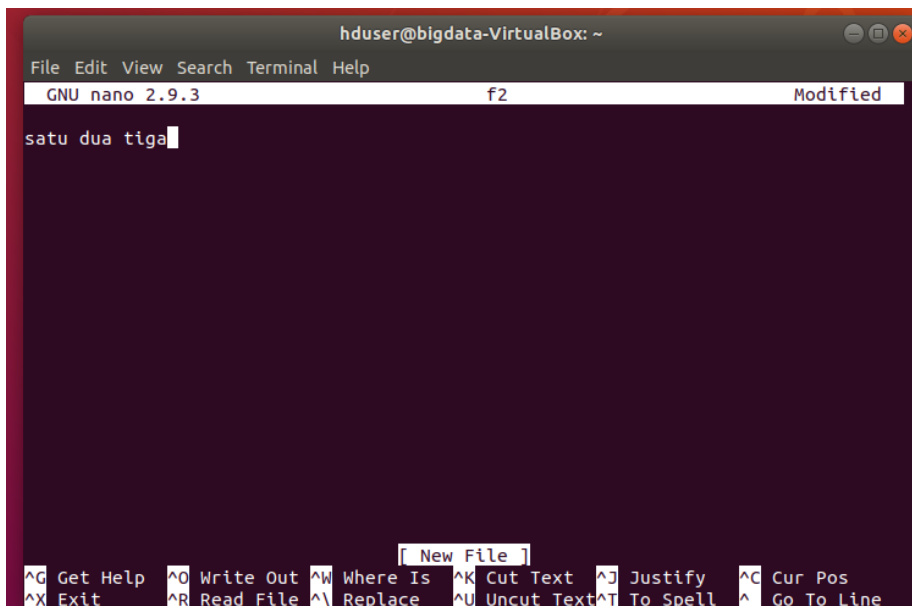
```
hduser@bigdata-VirtualBox: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 f1 Modified  
satu dua dua tiga tiga tiga  
  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Keluar dan simpan dengan Ctrl + X kemudian klik Y dan tekan ENTER.

12. Kemudian buat berkas/file baru lagi dengan nama f2 menggunakan perintah
sudo nano f2

Dalam berkas/file f2 tersebut, isi dengan teks di bawah ini sesuai gambar yang disediakan

satu dua tiga



```
hduser@bigdata-VirtualBox: ~  
File Edit View Search Terminal Help  
GNU nano 2.9.3 f2 Modified  
satu dua tiga  
  
[ New File ]  
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos  
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Keluar dan simpan dengan Ctrl + X kemudian klik Y dan tekan ENTER.

Kedua berkas yang baru saja dibuat ini nantinya yang akan dijadikan objek penghitungan kata oleh Hadoop.

13. Periksa apakah berkas sudah ada dalam folder menggunakan perintah

ls

```
hduser@bigdata-VirtualBox:~$ ls
examples.desktop  textkopi.txt      'WordCount$IntSumReducer.class'
f1                text.txt          WordCount.java
f2                tmpdata           'WordCount$TokenizerMapper.class'
hadoop-3.3.0      wc.jar
hadoop-3.3.0.tar.gz WordCount.class
```

14. Periksa apakah YARN dan Hadoop sudah berjalan dengan perintah

jps

```
hduser@bigdata-VirtualBox:~$ jps
1992 Jps
hduser@bigdata-VirtualBox:~$
```

Jika hasil masih seperti pada gambar diatas (hanya berupa Jps), berarti YARN dan Hadoop belum berjalan. Jika sudah ada, maka bisa lanjut ke langkah berikutnya. Jika belum ada, maka jalankan perintah berikut

```
~/hadoop-3.3.0/sbin/start-all.sh
```

Hasilnya seharusnya keluar seperti gambar dibawah ini.

```
hduser@bigdata-VirtualBox:~$ ~/hadoop-3.3.0/sbin/start-all.sh
WARNING: Attempting to start all Apache Hadoop daemons as hduser in 10 seconds.
WARNING: This is not a recommended production deployment configuration.
WARNING: Use CTRL-C to abort.
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [bigdata-VirtualBox]
Starting resourcemanager
Starting nodemanagers
hduser@bigdata-VirtualBox:~$
```

Kemudian periksa kembali dengan jps

```
hduser@bigdata-VirtualBox:~$ jps
2512 DataNode
3251 NodeManager
3637 Jps
2346 NameNode
3083 ResourceManager
2748 SecondaryNameNode
hduser@bigdata-VirtualBox:~$
```

Jika hasil sesuai seperti gambar di atas, maka YARN dan Hadoop sudah berjalan, kemudian dapat lanjut ke langkah berikutnya.

15. Masukkan kedua berkas f1 dan f2 ke dalam hdfs dengan alamat /user/modul7/input dengan perintah

```
hadoop fs -copyFromLocal f1 /user/modul7/input
```

```
hadoop fs -copyFromLocal f2 /user/modul7/input
```

```
hduser@bigdata-VirtualBox:~$ hadoop fs -copyFromLocal f1 /user/modul7/input
hduser@bigdata-VirtualBox:~$ hadoop fs -copyFromLocal f2 /user/modul7/input
```

16. Periksa apakah berkas sudah masuk dan tersedia dengan perintah

```
hadoop fs -ls /user/modul7/input
```

```
hduser@bigdata-VirtualBox:~$ hadoop fs -ls /user/modul7/input
Found 2 items
-rw-r--r--  1 hduser supergroup      28 2020-11-14 16:19 /user/modul7/input/
f1
-rw-r--r--  1 hduser supergroup      14 2020-11-14 16:20 /user/modul7/input/
f2
```

17. Jalankan file jar tersebut untuk menghitung jumlah kata pada file input yang ada di folder *input* pada HDFS, kemudian simpan hasilnya di folder hasil

```
hadoop jar wc.jar WordCount /user/modul7/input
/user/modul7/output/hasil
```



```

hduser@bigdata-VirtualBox:~$ hadoop jar wc.jar WordCount /user/modul7/input /user/modul7/output/hasil
2020-11-15 11:20:03,201 INFO client.DefaultNoHARMFalloverProxyProvider: Connecting to ResourceManager at /127.0.0.1:8032
2020-11-15 11:20:03,694 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2020-11-15 11:20:03,776 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/hduser/.staging/job_1605413742844_0001
2020-11-15 11:20:04,786 INFO input.FileInputFormat: Total input files to process : 2
2020-11-15 11:20:05,083 INFO mapreduce.JobSubmitter: number of splits:2
2020-11-15 11:20:05,741 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1605413742844_0001
2020-11-15 11:20:05,741 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-11-15 11:20:06,083 INFO conf.Configuration: resource-types.xml not found
2020-11-15 11:20:06,084 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-11-15 11:20:06,486 INFO impl.YarnClientImpl: Submitted application application_1605413742844_0001
2020-11-15 11:20:06,602 INFO mapreduce.Job: The url to track the job: http://bigdata-VirtualBox:8088/proxy/application_1605413742844_0001/
2020-11-15 11:20:06,602 INFO mapreduce.Job: Running job: job_1605413742844_0001
2020-11-15 11:20:19,073 INFO mapreduce.Job: Job job_1605413742844_0001 running in uber mode : false
2020-11-15 11:20:19,077 INFO mapreduce.Job: map 0% reduce 0%
2020-11-15 11:20:49,058 INFO mapreduce.Job: map 100% reduce 0%
2020-11-15 11:21:38,153 INFO mapreduce.Job: map 100% reduce 100%
2020-11-15 11:21:43,516 INFO mapreduce.Job: Job job_1605413742844_0001 completed successfully
2020-11-15 11:21:44,729 INFO mapreduce.Job: Counters: 54

```

18. Periksa aplikasi yang berjalan pada YARN dengan masuk pada web browser, kemudian ketik alamat <http://localhost:8088/cluster>



▼ Cluster

- About
- Nodes
- Node Labels
- Applications
 - NEW
 - NEW SAVING
 - SUBMITTED
 - ACCEPTED
 - RUNNING
 - FINISHED
 - FAILED
 - KILLED
- Scheduler

► Tools

Cluster Metrics

Apps Submitted	Apps Pending	Apps Failed
1	0	0

Cluster Nodes Metrics

Active Nodes	Decommissioning Nodes
1	0

Scheduler Metrics

Scheduler Type	Capacity
Capacity Scheduler	[memory-mb (unit=M), vcores]

Show 20 entries

ID	User	Name	Application Type
application_1605413742844_0001	hduser	word count	MAPREDUCE

Showing 1 to 1 of 1 entries

Kemudian kembali ke terminal dan hentikan proses dengan menjalankan perintah `ctrl+c`.

19. Lakukan pengecekan apakah hasil perhitungan kata telah berhasil masuk ke dalam direktori hasil menggunakan perintah
- ```
hadoop fs -ls /user/modul7/output/hasil
```

```
bytes written=20
hduser@bigdata-VirtualBox:~$ hadoop fs -ls /user/modul7/output/hasil
Found 2 items
-rw-r--r-- 1 hduser supergroup 0 2020-11-15 11:21 /user/modul7/output/hasil/_SUCCESS
-rw-r--r-- 1 hduser supergroup 20 2020-11-15 11:21 /user/modul7/output/hasil/part-r-000000
```

Pada folder hasil terdapat dua *file* seperti gambar di atas. Hasil perhitungan bisa dilihat pada file yang nama depan part, pada kasus ini part-r-000000.

## 20. Melihat hasil perhitungan Wordcount di file part-r-000000

```
hadoop fs -cat /user/modul7/output/hasil/part-r-000000
```

```
hduser@bigdata-VirtualBox:~$ hadoop fs -cat /user/modul7/output/hasil/part-r-000000
dua 3
satu 2
tiga 4
```

---