MAP REDUCE PATTERN

Arranged By:

Rajendra Rakha Arya Prabaswara

(1941720080/20)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG

1. Copy the dataset into the Hadoop directory

2. Create a new folder inside hadoop server

3. Move City.csv and Temperature.csv to a new folder

4. Full Code App.java

```java
package org.example;

import
org.apache.hadoop.conf.Configur
ation;
import
org.apache.hadoop.fs.Path;
import
org.apache.hadoop.io.IntWritabl
e;
import
org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapreduce.Job
;
import
org.apache.hadoop.mapreduce.Map
per;
```

```java
import
org.apache.hadoop.mapreduce.Red
ucer;
import
org.apache.hadoop.mapreduce.lib
.input.MultipleInputs;
import
org.apache.hadoop.mapreduce.lib
.input.TextInputFormat;
import
org.apache.hadoop.mapreduce.lib
.output.FileOutputFormat;
import java.io.IOException;

public class App
{
    public static void main(
String[] args ) throws
Exception{
```

```
        Configuration conf =
  new Configuration();
        Job job =  new
Job(conf, "City Temperature
Job");

job.setMapperClass(TemperatureM
apper.class);
        System.out.println(new
Path(args[0]));
        System.out.println(new
Path(args[1]));

MultipleInputs.addInputPath(job
, new Path(args[0]),

TextInputFormat.class,
CityMapper.class);

MultipleInputs.addInputPath(job
, new Path(args[1]),

TextInputFormat.class,
TemperatureMapper.class);
```

```
job.setMapOutputKeyClass(Text.c
lass);

job.setMapOutputValueClass(Text
.class);

job.setReducerClass(Temperature
Reducer.class);

job.setOutputKeyClass(Text.clas
s);

job.setOutputValueClass(IntWrit
able.class);

FileOutputFormat.setOutputPath(
job, new Path(args[2]));

System.exit(job.waitForCompleti
on(true) ? 0 : 1);
    }
}
```

5. Full Code CityMapper.java

```
package org.example;

import
org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapreduce.Map
per;

import java.io.IOException;

public class CityMapper
        extends Mapper<Object,
Text, Text, Text> {
    public void map(Object key,
Text value, Context context)
            throws IOException,
InterruptedException{
```

```
        String txt =
value.toString();
        String[] tokens =
txt.split(",");
        String id =
tokens[0].trim();
        String name =
tokens[1].trim();
        if
(name.compareTo("City") != 0)
            context.write(new
Text(id), new Text(name));

    }
}
```

6. Full Code TemperatureMapper.java

```
package org.example;

import
org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapreduce.Map
per;

import java.io.IOException;

public class TemperatureMapper
        extends Mapper<Object,
Text, Text, Text> {
    public void map (Object
key, Text value, Context
context)
            throws IOException,
InterruptedException{
```

```
        String txt =
value.toString();
        String[] tokens =
txt.split(",");
        String date =
tokens[0];
        String id =
tokens[1].trim();
        String temperature =
tokens[2].trim();
        if
(temperature.compareTo("Tempera
ture") != 0 )
            context.write(new
Text(id), new
Text(temperature));
    }
}
```

7. Full Code TemperatureReducer.java

```java
package org.example;

import
org.apache.hadoop.io.IntWritabl
e;
import
org.apache.hadoop.io.Text;
import
org.apache.hadoop.mapreduce.Red
ucer;

import java.io.IOException;

public class TemperatureReducer
        extends Reducer<Text,
Text, Text, IntWritable> {
    private final IntWritable
result = new IntWritable();
    private Text cityName = new
Text( "Unknown");

    public void reduce(Text
key, Iterable<Text> values,
                       Context
context) throws IOException,
InterruptedException {
        int sum = 0;
        int n = 0;
        cityName = new
Text("city" + key.toString());
```

```java
        for (Text val : values)
{
            String strVal =
val.toString();
            if (strVal.length()
<= 3) {
                sum +=
Integer.parseInt(strVal);
                n += 1;
            } else {
                cityName = new
Text(strVal);
            }
        }


        if (n != 0 &&
cityName.toString().compareTo("
Unknown") != 0) {
            result.set(sum /
n);

context.write(cityName,
result);


        }
    }
}
```

8. Fill in run-job.sh

```bash
# Variabel-variabel
FILE_JAR=target/joinPattern-
1.0-SNAPSHOT.jar
HADOOP_USERNAME=hadoopuser
IP_NAMENODE=192.168.0.25
HOME_DIR=/home/hadoopuser/
NAMA_JAR_TUJUAN=joinPattern.jar
PACKAGE_ID=org.example.App
INPUT_FOLDER_CITY=/hadoopuser/J
oin/City
INPUT_FOLDER_TEMPERATURE=/hadoo
puser/Join/Temperature
OUTPUT_FOLDER=/hadoopuser/Join/
Hasil

# Bersihkan console
clear

# Jalankan SCP
SCP_ARG="${HADOOP_USERNAME}@${I
P_NAMENODE}:${HOME_DIR}${NAMA_J
AR_TUJUAN}"
echo "Running SCP.."
echo "${SCP_ARG}"
scp $FILE_JAR $SCP_ARG

# SSH ke Name Node dan jalankan
MapReduce job
```

```bash
echo "Connecting to Namenode
and execute MapReduce Job.."
HADOOP_JAR_COMMAND="hadoop jar
${NAMA_JAR_TUJUAN}
${PACKAGE_ID}
${INPUT_FOLDER_CITY}
${INPUT_FOLDER_TEMPERATURE}
${OUTPUT_FOLDER}"
LS_OUTPUT_COMMAND="hadoop fs -
ls ${OUTPUT_FOLDER}"
CAT_OUTPUT_COMMAND="hadoop fs -
cat ${OUTPUT_FOLDER}/part-r-
00000"
DELETE_OUTPUT_COMMAND="hadoop
fs -rm -r ${OUTPUT_FOLDER}"
ssh
"${HADOOP_USERNAME}@${IP_NAMENO
DE}" "${HADOOP_JAR_COMMAND};
${LS_OUTPUT_COMMAND};
${CAT_OUTPUT_COMMAND};
${DELETE_OUTPUT_COMMAND}; exit"
echo "Selesai."
```

9. Run run-job.sh

```
Found 2 items
-rw-r--r--   2 hadoopuser supergroup          0 2022-06-15 13:06 /hadoopuser/Join/Hasil/_SUCCESS
-rw-r--r--   2 hadoopuser supergroup         75 2022-06-15 13:06 /hadoopuser/Join/Hasil/part-r-00000
Boston  22
New York        23
Chicago 24
Philadelphia    24
San Francisco   25
city6   22
Deleted /hadoopuser/Join/Hasil
Selesai.
```

FULL CODE

https://github.com/Rjndrkha/JoinPattern.git