LAPORAN TUGAS KTP
MULTI NODE CLUSTER HADOOP

Disusun Oleh:

Andira Faqih Muhammad - 24060120140142
Julius Adrian - 24060120130127
Sabdiel Tarigan - 24060120120028

PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN MATEMATIKA
UNIVERSITAS DIPONEGORO
SEMARANG
2022

Langkah-Langkah Pembuatan Cluster Hadoop

1. Membuat virtual machine linux dengan OS Ubuntu/Lubuntu
2. Menginstall SSH pada virtual machine
3. Menginstall PDSH pada virtual machine
4. Mengedit/write file .bashrc dan memberi perintah
   `export PDSH_RCMD_TYPE=ssh` untuk mengekspor PDSH dan SSH yang sudah diinstall
5. Membuat SSH key baru dengan perintah
   `ssh-keygen -t rsa -P ""` dan memberikan konfirmasi bila ditanya
6. Mengcopy SSH key yang sudah dibuat ke dalam path authorized_keys dengan perintah
   `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
7. Lakukan verifikasi SSH terhadap localhost dengan perintah
   `ssh localhost`
8. Lakukan instalasi Java Development Kit (JDK) dengan perintah
   `sudo apt install openjdk-8-jdk`
9. Unduh hadoop di link mirrors.sonic.net dengan perintah
   `sudo wget -P ~ https://mirrors.sonic.net/apache/hadoop/common/hadoop-3.3.2/hadoop-3.3.2.tar.gz`
10. Karena unduhan di langkah 9 berupa zip, maka alangkah baiknya untuk di unzip terlebih dahulu menggunakan perintah
    `tar xzf hadoop-3.3.2.tar.gz`
11. Ubah nama folder yang masih menggunakan nama versi menjadi lebih simple, lakukan dengan perintah
    `mv hadoop-3.3.2 hadoop`
12. Buka hadoop-env.sh dan edit JAVA_HOME dengan menambahkan
    `export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/`
13. Ubah directory folder hadoop dengan perintah
    `sudo mv hadoop /usr/local/hadoop`
14. Buka file environment dan tambahkan path berikut :
    `PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/local/hadoop/bin:/usr/local/hadoop/sbin"JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64/jre"`
15. Buat user baru untuk hadoopnya dengan perintah
    `sudo adduser hadoopuser`
    Lalu tambahkan perintah-perintah berikut:
    `sudo usermod -aG hadoopuser hadoopuser`
    `sudo chown hadoopuser:root -R /usr/local/hadoop/`
    `sudo chmod g+rwx -R /usr/local/hadoop/`
    `sudo adduser hadoopuser sudo`
16. Check IP Address dengan perintah

`ip addr`

17. Buka file host dengan
    `sudo nano /etc/hosts`
    Dan tulis IP Address yang akan dipakai oleh master, slave 1, dan slave2
18. Matikan VM master dan lakukan clone sebanyak 2 kali, berikan nama slave1 dan slave 2 untuk hasil clone nya
19. Buka file hostname dan masukkan nama dari virtual machinenya. Bila sudah, lakukan reboot dengan perintah
    `sudo reboot`
20. Konfigurasikan SSH-nya hadoop-master dengan perintah
    `su - hadoopuser`
21. Buat lagi SSH key dengan perintah
    `ssh-keygen -t rsa`
22. Kemudian kita salin SSH ke semua user dengan perintah
    `ssh-copy-id hadoopuser@hadoop-master`
    `ssh-copy-id hadoopuser@hadoop-slave1`
    `ssh-copy-id hadoopuser@hadoop-slave2`
23. Gunakan nano untuk membuka core.site.xml dan masukkan baris-baris berikut
    ```
    <configuration>
    <property>
    <name>fs.defaultFS</name>
    <value>hdfs://hadoop-master:9000</value>
    </property>
    </configuration>
    ```
24. Lalu buka hdfs-site.xml dan masukkan baris-baris berikut
    ```
    <configuration>
    <property>
    <name>dfs.namenode.name.dir</name><value>/usr/local/hadoo
    p/data/nameNode</value>
    </property>
    <property>
    <name>dfs.datanode.data.dir</name><value>/usr/local/hadoo
    p/data/dataNode</value>
    </property>
    <property>
    <name>dfs.replication</name>
    <value>2</value>
    </property>
    </configuration>
    ```
25. Buka file workers dan salin kedua baris berikut
    ```
    hadoop-slave1
    hadoop-slave2
    ```
26. Tiru konfigurasi master kepada slave-slavenya dengan perintah

```
scp /usr/local/hadoop/etc/hadoop/*
hadoop-slave1:/usr/local/hadoop/etc/hadoop/
scp /usr/local/hadoop/etc/hadoop/*
hadoop-slave2:/usr/local/hadoop/etc/hadoop/
```

27. Format Hadoop File Systemnya dengan perintah
    ```
    source /etc/environment
    hdfs namenode -format
    ```
28. Jalankan Hadoop File System dengan perintah
    ```
    start-dfs.sh
    ```
    Lalu cek dengan perintah dibawah untuk melihat resource mana saja yang diinisialisasikan
    ```
    jps
    ```
29. Cek di browser hadoop-master:9870 untuk melihat apakah hadoopnya sudah berjalan
30. Konfigurasikan yarn dengan perintah berikut
    ```
    export HADOOP_HOME="/usr/local/hadoop"
    export HADOOP_COMMON_HOME=$HADOOP_HOME
    export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
    export HADOOP_HDFS_HOME=$HADOOP_HOME
    export HADOOP_MAPRED_HOME=$HADOOP_HOME
    export HADOOP_YARN_HOME=$HADOOP_HOME
    ```
31. Pada kedua slave, buka yarn-site.xml dan tambahkan perintah berikut
    ```
    <property>
    <name>yarn.resourcemanager.hostname</name>
    <value>hadoop-master</value>
    </property>
    ```
32. Jalankan yarn dengan perintah di bawah
    ```
    start-yarn.sh
    ```
33. Masuk ke browser dan buka http://hadoop-master:8088/cluster

Penjelasan Dataset dan Program

Data set .txt dengan jumlah kata 1.159.047
Link file txt:
https://drive.google.com/file/d/1FVMzJ9Wbvyw5cZEMwPH5czueQH9p3J0x/view?usp=sharing

Pada hadoop sudah ada program word count untuk menghitung jumlah kata yang sama pada file, cara untuk meng-eksekusinya adalah sebagai berikut :

1. Menjalankan Hadoop
   ```
   start-dfs.sh
   start-yarn.sh
   ```

2. Membuat direktori dengan nama **input12**
   ```
   hadoop fs -mkdir/input12
   ```

3. Membuat file inputWordCount.txt, dan mengisikan dengan kata kata yang akan dihitung jumlah kata yang sama.
   ```
   sudo nano inputWordCount.txt
   ```

4. memindahkan file inputWordCount.txt yang berisi kata kata kedalam direktori input12.
   ```
   hadoop fs -put inputWordCount.txt/input12
   ```

5. eksekusi file
   ```
   hadoop jar WordCount.jar
   WordCount/input12/inputWordCount.txt/WordCount-Result12
   ```

6. cek folder word-count-result12
   ```
   hadoop fs -ls/WordCount-Result12
   ```

7. melihat hasil perhitungan word count pada file part-r-00000
   ```
   hadoop fs -cat/WordCount-Result12/part-r-00000
   ```

```
import java.io.IOException;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
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 {
   // Map function
public static class MyMapper extends Mapper<LongWritable, Text, Text, IntWritable>{
      private Text word = new Text();
      public void map(LongWritable key, Text value, Context context)
            throws IOException, InterruptedException {
         // Splitting the line on spaces
         String[] stringArr = value.toString().split("\\s+");
         for (String str : stringArr) {
            word.set(str);
            context.write(word, new IntWritable(1));
         }
      }
   }

   // Reduce function
   public static class MyReducer extends Reducer<Text, IntWritable, Text, IntWritable>{
      private IntWritable result = new IntWritable();
```

```
    public void reduce(Text key, Iterable<IntWritable> values, Context context)
          throws IOException, InterruptedException {
      int sum = 0;
      for (IntWritable val : values) {
        sum += val.get();
      }
      result.set(sum);
      context.write(key, result);
    }
  }
  public static void main(String[] args)  throws Exception{
    Configuration conf = new Configuration();

    Job job = Job.getInstance(conf, "WC");
    job.setJarByClass(WordCount.class);
    job.setMapperClass(MyMapper.class);
    job.setReducerClass(MyReducer.class);
    job.setOutputKeyClass(Text.class);
    job.setOutputValueClass(IntWritable.class);
    FileInputFormat.addInputPath(job, new Path(args[0]));
    FileOutputFormat.setOutputPath(job, new Path(args[1]));
    System.exit(job.waitForCompletion(true) ? 0 : 1);
  }
}
```

- Class MyMapper melakukan extend pada class Mapper yang telah didefinisikan dalam MapReduce
- Input dan Output Mapper merupakan pasangan key dan value, untuk bagian input,key pada program tersebut adalah LongWritable dan value dari program tersebut adalah Text. Untuk bagian output, key adalah Text, value adalah IntWritable
- Class MyReducer melakukan extend pada class Reducer yang telah didefinisikan pada MapReduce
- Input dan Output reducer merupakan pasangan key dan value, pada bagian input, key pada program tersebut adalah Text, value dari program tersebut adalah IntWritable. pada bagian Output, key pada program tersebut adalah Text, dan Value dari program tersebut adalah IntWritable.
- Pada class driver, dilakukan konfigurasi MapReduce untuk dijalankan pada Hadoop
- Menentukan job,tipe data input output mapper dan reducer.
- Menentukan nama class pada mapper dan reducer.
- Menentukan path folder input output.
- Function setInputFormatClass() berfungsi untuk bagaimana mapper akan membaca input. Memilih TextInputFormat sehingga baris kata akan dibaca oleh mapper dalam satu waktu
- Dalam fungsi main dibuat instance objek konfigurasi baru.

Cara Eksekusi Wordcount

1. Mengcompile file WordCount.java yang telah dibuat
   ```
   javac -classpath $HADOOP_CLASSPATH -d WordCountCompiled/
   WordCount.java
   ```

2. Mengubah menjadi file executable.jar
   ```
   jar -cvf WordCount.jar -C WordCountCompiled/ .
   ```

3. Menjalankan file jar tersebut untuk menghitung jumlah kata pada file *inputWordCount.txt* yang ada di folder *input2* pada HDFS.

4. Menyimpan hasilnya di folder *WordCount-Result* menggunakan command
   ```
   hadoop jar WordCount.jar WordCount /input/inputWordCount.
   txt /WordCount-Result
   ```

Hasil Eksekusi

```
hadoopuser@hadoop-master:~$ hadoop jar WordCount.jar WordCount /input3/inputWordCount.txt /WordCoun
t-Result3
2022-10-18 18:39:32,179 INFO impl.MetricsConfig: Loaded properties from hadoop-metrics2.properties
2022-10-18 18:39:32,314 INFO impl.MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(
s).
2022-10-18 18:39:32,315 INFO impl.MetricsSystemImpl: JobTracker metrics system started
2022-10-18 18:39:32,528 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not
performed. Implement the Tool interface and execute your application with ToolRunner to remedy this
.
2022-10-18 18:39:32,667 INFO input.FileInputFormat: Total input files to process : 1
2022-10-18 18:39:32,757 INFO mapreduce.JobSubmitter: number of splits:1
2022-10-18 18:39:33,078 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_local1968356576
_0001
2022-10-18 18:39:33,084 INFO mapreduce.JobSubmitter: Executing with tokens: []
2022-10-18 18:39:33,347 INFO mapreduce.Job: The url to track the job: http://localhost:8080/
2022-10-18 18:39:33,349 INFO mapreduce.Job: Running job: job_local1968356576_0001
2022-10-18 18:39:33,354 INFO mapred.LocalJobRunner: OutputCommitter set in config null
2022-10-18 18:39:33,375 INFO output.FileOutputCommitter: File Output Committer Algorithm version is
 2
2022-10-18 18:39:33,382 INFO output.FileOutputCommitter: FileOutputCommitter skip cleanup _temporar
y folders under output directory:false, ignore cleanup failures: false
2022-10-18 18:39:33,383 INFO mapred.LocalJobRunner: OutputCommitter is org.apache.hadoop.mapreduce.
lib.output.FileOutputCommitter
2022-10-18 18:39:33,478 INFO mapred.LocalJobRunner: Waiting for map tasks
2022-10-18 18:39:33,478 INFO mapred.LocalJobRunner: Starting task: attempt_local1968356576_0001_m_0
00000_0
2022-10-18 18:39:33,526 INFO output.FileOutputCommitter: File Output Committer Algorithm version is
 2
```

```
hadoopuser@hadoop-master:~$ hadoop fs -ls /WordCount-Result3
Found 2 items
-rw-r--r--   2 hadoopuser supergroup          0 2022-10-18 18:39 /WordCount-Result3/_SUCCESS
-rw-r--r--   2 hadoopuser supergroup       4381 2022-10-18 18:39 /WordCount-Result3/part-r-00000
hadoopuser@hadoop-master:~$ hadoop fs -cat /WordCount-Result3/part-r-00000
2022-10-18 18:39:56,924 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTru
sted = false, remoteHostTrusted = false
        3
"Getting        5
"God    5
"How    5
"Oh,    5
"What's 5
"it     5
"what   5
-       20
A       5
And     19
But     9
Doing   5
Drops   5
First   5
For     5
God",   5
Gregor  18
Gregor's        5
Had     5
He      25
He'd    5
Heaven!"        5
Hell!"  5
His     14
```

Berikut adalah hasil lengkapnya:

	3
"Getting	5
"God	5
"How	5
"Oh,	5
"What's	5
"it	5
"what	5
-	20
A	5
And	19
But	9
Doing	5
Drops	5
First	5
For	5
God",	5
Gregor	18
Gregor's	5
Had	5
He	25
He'd	5
Heaven!"	5

Hell!" 5

His 14

However 5

I 40

I'd 15

I'll 10

I've 15

If 5

It 20

One 4

Other 5

Samsa 10

That's 5

The 15

Travelling 5

True, 5

Well, 5

What 9

Yes, 5

You've 5

a 109

able 5

about 29

above 5

accept 4

accuse 4

after 4

ago, 5

ago. 5

alarm 10

all 35

although 5

always 10

an 5

and 112

anger 5

another 5

any 9

anyone 5

apart 4

arches 5

are 5

arm 5

armour-like 5

as 33

assistant 10

at 45

avoid 5

back 25

back, 5

bad 5

be 19

because 20

become 5

bed 10

bedding 5

been 18

been; 5

before. 5

began 5

being 5

believed 4

belly, 5

belly; 5

best 5

better; 5

between 5

big 5

bit 5

boa 5

boss 14

boss's 10

boss; 5

breakfasts. 5

brown 5

business 15

but 19

by 10

can 10

career 5

case? 4

catch 10

certainly 9

change. 5

chest 5

chosen! 5

claim 4

clock 5

clock, 5

close 5

cold 5

collection 10

come 4

company, 4

compared 5

completely 4

connections, 5

contact 5

contract, 5

copy 5

could 20

couldn't 5

cover 5

covered 10

curse 5

cut 5

day 10

debt 5

deeply 5

definitely 5

desk! 5

desk, 5

did 14

didn't 10

different 5

divided 5

do 10

do. 5

doctor 8

doctor's 4

doing 5

domed 5

down 5

drawers. 5

dream. 5

dreams, 5

drew 5

dull 10

during 5

early 5

eating 5

effort 5

enough 5

entirely 4

especially 5

even 14

ever 4

everything 5

excessive 4

extremely 4

eyes 5

fact, 4

fall 5

familiar 5

feel 24

feel. 5

felt 14

fifteen 4

fitted 5

five 10

five." 5

floundering 5

food, 5

for 14

forget 5

former 5

forwards, 5

found 10

four 10

frame. 5

fresh 5

friendly 5

from 23

funny 5

fur 15

furniture-rattling 5

gentlemen 5

get 25

gilded 5

given 5

go 15

go, 5

gone 5

got 15

guest 5

had 24

half 10

hands 5

happened 5

hard 10

hardly 5

hat 5

have 59

having 4

he	173
head	10
headboard	5
heard	5
hearing.	5
heavy	5
helplessly	5
her	5
him	25
him,	5
himself	15
his	84
hitting	5
home,	5
hope;	5
horrible	5
house	5
housed	5
human	5
hundred	5
hung	5
hungrier	4
if	24

ill	4
ill.	4
illustrated	5
in	47
instance,	5
insurance	4
into	20
irregular	5
is	10
it	60
it's	5
itch	10
its	5
just	15
kicked	5
know	15
knows,	5
lady	5
later	5
lay	15
lazy	4
leaves	5
legs	5

legs,	10
let	5
life	5
lift	5
lifted	5
like	15
little	20
live	5
lively.	5
long	10
long,	4
longer	5
look	10
looked	5
looked.	5
lots	5
lower	5
luxury.	5
mad	5
made	5
magazine	5
make	14
makes	5

making 5

man, 5

many 9

maybe 5

me. 5

me?" 5

medical 4

mild, 5

moment. 5

money 5

more 15

more, 4

morning 5

morning, 5

moving 5

much 9

muff 5

must 10

my 25

never 14

next 5

nice, 5

no 4

no-one 4

noise? 5

nonsense", 5

not 34

notice 5

now? 5

o'clock 10

of 93

of; 5

off 15

office 10

on 40

once 9

one 5

only 5

onto 5

or 10

ought 5

out 30

out. 5

over 5

overcome 5

own 5

packed, 5

pain 5

pane, 5

parents 9

parents' 5

particularly 5

past 5

past, 5

pay 5

peacefully 5

peacefully, 5

people 5

picture 5

pitifully 5

place 5

position. 10

possible 5

present 5

probably 5

proper 5

pushed 5

put 5

quarter 5

quickly 5

quietly 10

quite 5

rain 5

raising 5

ready 5

recently 5

recommendation 4

report 5

reported 4

rest 5

right 10

right, 10

rolled 5

room 5

room, 5

round 4

rung. 5

rung? 5

rush 5

sad. 5

salesman 5

salesmen 5

samples 10

sat 5

saw 5

sections. 5

see 15

seemed 5

service 4

set 5

seven. 5

seven; 5

should 10

showed 5

shudder. 5

shut 5

sick? 4

sitting 10

six 10

size 5

sleep 10

sleep. 5

sleepiness 4

sleeping 9

slept 5

slid 5

slide 5

slight 5

slightly 5

slowly 5

small, 5

so 19

some 5

something 5

son, 4

soon 5

sort 5

spineless, 5

spot. 5

spots 5

spread 5

state 5

stiff 5

still 15

stopped 5

strained 4

strenuous 5

stupid. 5

subordinates 5

suppose 5

suspicious 4

table 5

takes 5

talking 5

tell 5

textile 5

than 14

that 92

that's 5

that. 5

the 191

their 5

them. 5

then 5

there 30

there's 10

there, 5

these 5

thin 5

thing 5

think 5

think, 5

this 14

though, 5

thought, 15

thought. 10

threw 5

through 5

ticking 5

time 15

time", 5

times, 5

to 149

together 5

told 5

too 5

top 5

touched 5

towards 10

train 25

transformed 5

travelling 10

travelling, 5

tried 10

troubled 5

try 5

turned 5

unable 5

understanding. 4

up 35

up, 5

upright, 5

used 5

usual. 3

usual.One 1

vermin. 5

viewer. 5

walls. 5

was 64

was, 5

was. 5

wasn't 5

waved 5

weather. 5

well 4

went 5

were 14

what 20

what's 4

when 25

whenever 5

where 10

which 10

white 5

who 10

whole 5

window 5

with 44

woke 5

workshy. 4

worries 5

would 37

would, 5

wouldn't 5

wrong 4

years 9

yet 4

you 15

your 15

Link Video Demo

https://drive.google.com/file/d/1uxnZ6_iM5yK5GPDaMGwkdcVKWrxo9ujL/view?usp=sharing