

Tugas 4 Big Data



Created By:

[Muhamad Al Kausar Ramadhan]

[2041720193]

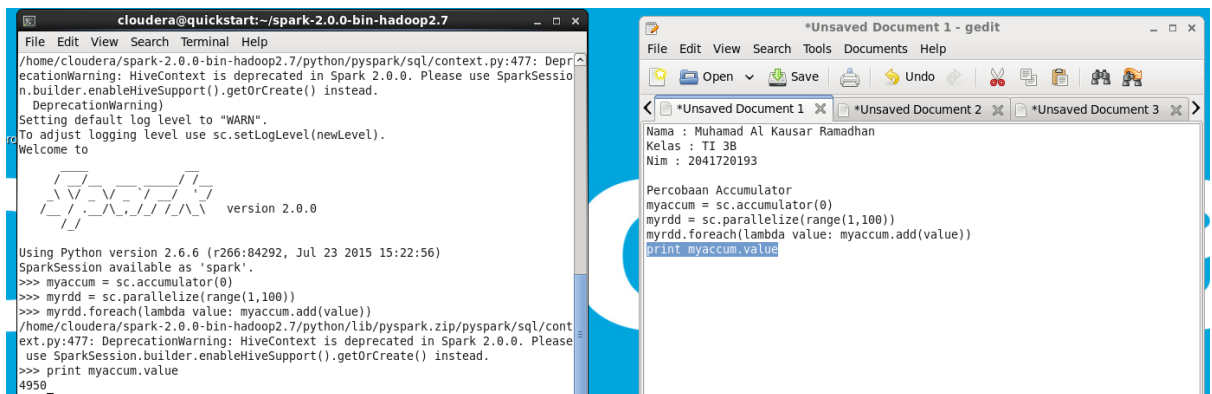
D4 TEKNIK INFORMATIKA

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2023

1. Accumulator



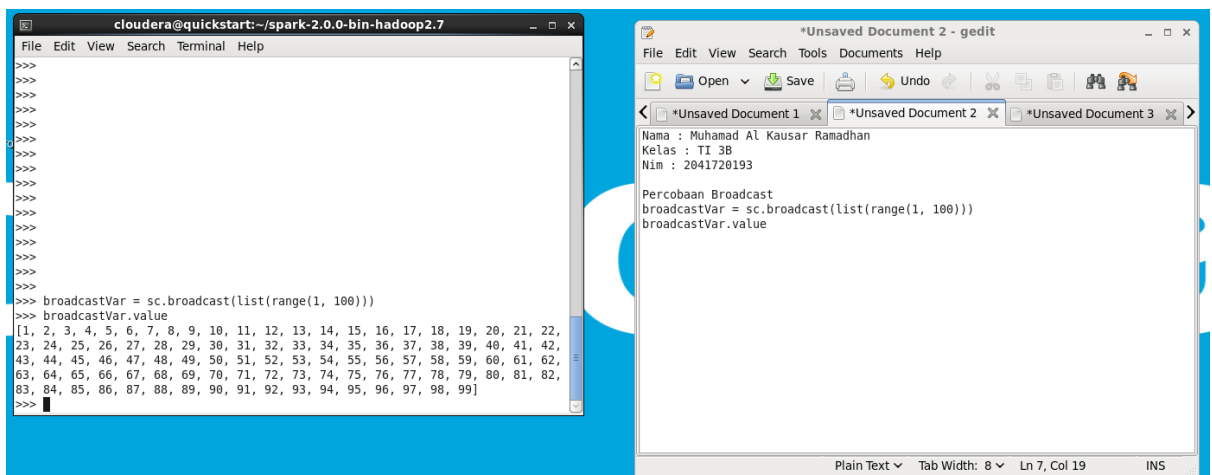
The screenshot shows a terminal window on the left and a text editor on the right. The terminal window, titled 'cloudera@quickstart:~/spark-2.0.0-bin-hadoop2.7', displays the Spark shell output. It shows the Spark version 2.0.0, the Python version 2.6.6, and the SparkSession available as 'spark'. The code in the terminal is as follows:

```
>>> myaccum = sc.accumulator(0)
>>> myrdd = sc.parallelize(range(1,100))
>>> myrdd.foreach(lambda value: myaccum.add(value))
>>> print myaccum.value
4950
```

The text editor, titled '*Unsaved Document 1 - gedit', shows the same code as the terminal, with the output '4950' printed at the end of the program.

Penjelasan: Kode program di atas menggunakan Apache Spark untuk melakukan parallel processing pada sebuah RDD (Resilient Distributed Dataset) dan menghitung jumlah total dari nilai-nilai dalam RDD tersebut.

2. Broadcast



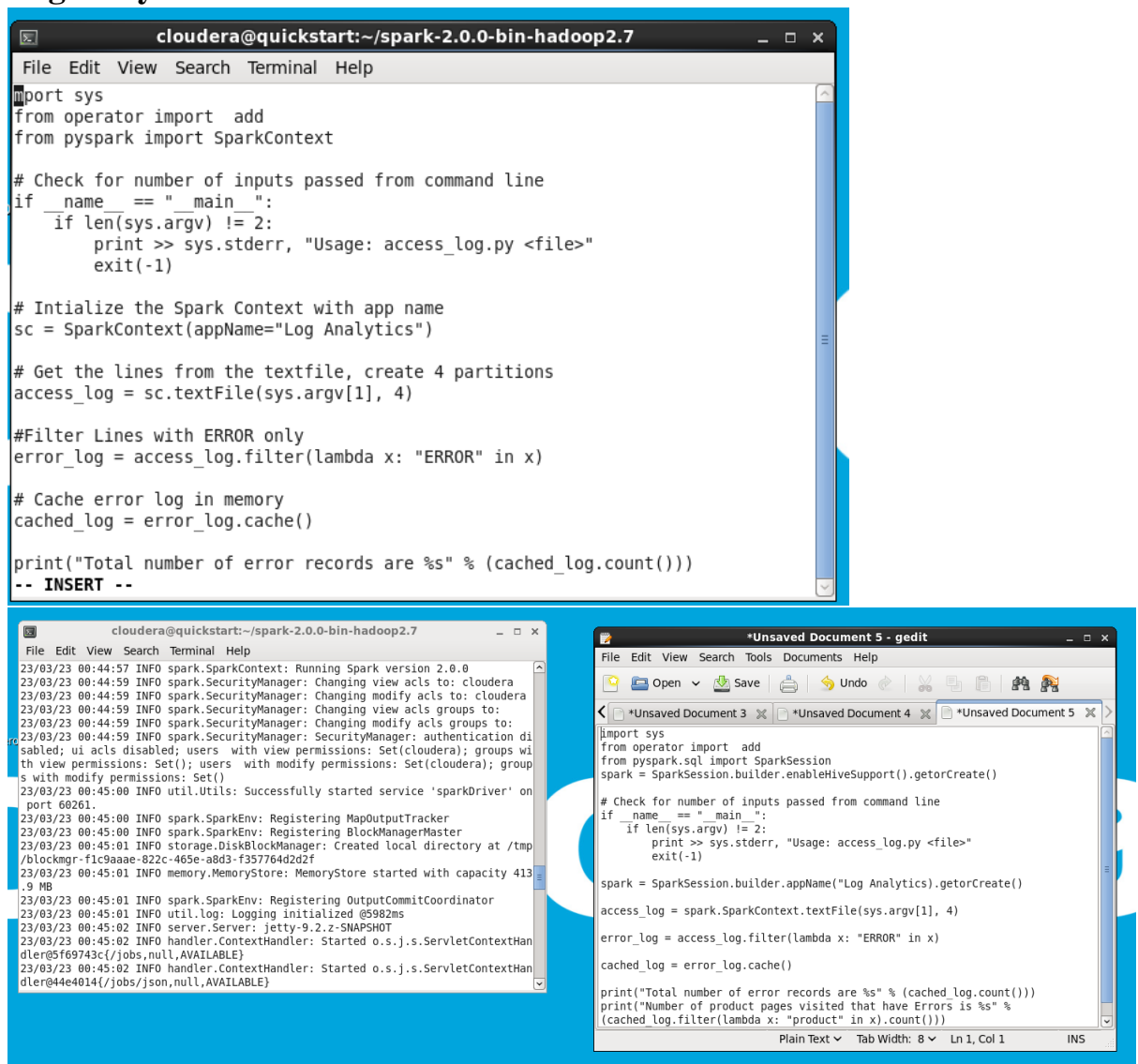
The screenshot shows a terminal window on the left and a text editor on the right. The terminal window, titled 'cloudera@quickstart:~/spark-2.0.0-bin-hadoop2.7', displays the Spark shell output. It shows the Spark version 2.0.0, the Python version 2.6.6, and the SparkSession available as 'spark'. The code in the terminal is as follows:

```
>>> broadcastVar = sc.broadcast(list(range(1, 100)))
>>> broadcastVar.value
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99]
```

The text editor, titled '*Unsaved Document 2 - gedit', shows the same code as the terminal, with the output of the broadcast variable printed at the end of the program.

Penjelasan: Kode program di atas menggunakan Apache Spark untuk membuat sebuah objek broadcast variable pada RDD (Resilient Distributed Dataset) yang berisi list nilai dari 1 hingga 99. Broadcast variable merupakan variabel yang dapat dibaca oleh semua worker nodes pada Spark cluster, dan digunakan untuk mengirimkan nilai yang sama ke setiap worker node secara efisien.

3. LogAnalytics



The image displays three screenshots related to the development and execution of a Spark log analysis program.

The top screenshot shows a terminal window titled `cloudera@quickstart:~/spark-2.0.0-bin-hadoop2.7` with a menu bar (File, Edit, View, Search, Terminal, Help). The code being written is as follows:

```
import sys
from operator import add
from pyspark import SparkContext

# Check for number of inputs passed from command line
if __name__ == "__main__":
    if len(sys.argv) != 2:
        print >> sys.stderr, "Usage: access_log.py <file>"
        exit(-1)

# Initialize the Spark Context with app name
sc = SparkContext(appName="Log Analytics")

# Get the lines from the textfile, create 4 partitions
access_log = sc.textFile(sys.argv[1], 4)

# Filter Lines with ERROR only
error_log = access_log.filter(lambda x: "ERROR" in x)

# Cache error log in memory
cached_log = error_log.cache()

print("Total number of error records are %s" % (cached_log.count()))
-- INSERT --
```

The bottom-left screenshot shows the same terminal window after execution. It displays a series of log messages from the Spark framework, including:

- `23/03/23 00:44:57 INFO spark.SparkContext: Running Spark version 2.0.0`
- `23/03/23 00:44:59 INFO spark.SecurityManager: Changing view acls to: cloudera`
- `23/03/23 00:44:59 INFO spark.SecurityManager: Changing modify acls to: cloudera`
- `23/03/23 00:44:59 INFO spark.SecurityManager: Changing view acls groups to:`
- `23/03/23 00:44:59 INFO spark.SecurityManager: Changing modify acls groups to:`
- `23/03/23 00:44:59 INFO spark.SecurityManager: SecurityManager: authentication disabled; ui acls disabled; users with view permissions: Set(cloudera); groups with view permissions: Set(); users with modify permissions: Set(cloudera); groups with modify permissions: Set()`
- `23/03/23 00:45:00 INFO util.Utils: Successfully started service 'sparkDriver' on port 60261.`
- `23/03/23 00:45:00 INFO spark.SparkEnv: Registering MapOutputTracker`
- `23/03/23 00:45:00 INFO spark.SparkEnv: Registering BlockManagerMaster`
- `23/03/23 00:45:01 INFO storage.DiskBlockManager: Created local directory at /tmp/blockmgr-f1c9aaae-822c-465e-a8d3-f357764d2d2f`
- `23/03/23 00:45:01 INFO memory.MemoryStore: MemoryStore started with capacity 413.9 MB`
- `23/03/23 00:45:01 INFO spark.SparkEnv: Registering OutputCommitCoordinator`
- `23/03/23 00:45:01 INFO util.log: Logging initialized 65982ms`
- `23/03/23 00:45:02 INFO server.Server: jetty-9.2.z-SNAPSHOT`
- `23/03/23 00:45:02 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@5f69743c[/jobs,null,AVAILABLE]`
- `23/03/23 00:45:02 INFO handler.ContextHandler: Started o.s.j.s.ServletContextHandler@44e4014[/jobs/json,null,AVAILABLE]`

The bottom-right screenshot shows a text editor window titled `*Unsaved Document 5 - gedit` with a menu bar (File, Edit, View, Search, Tools, Documents, Help). The code being edited is as follows:

```
import sys
from operator import add
from pyspark.sql import SparkSession
spark = SparkSession.builder.enableHiveSupport().getOrCreate()

# Check for number of inputs passed from command line
if __name__ == "__main__":
    if len(sys.argv) != 2:
        print >> sys.stderr, "Usage: access_log.py <file>"
        exit(-1)

spark = SparkSession.builder.appName("Log Analytics").getOrCreate()
access_log = spark.textFile(sys.argv[1], 4)
error_log = access_log.filter(lambda x: "ERROR" in x)
cached_log = error_log.cache()

print("Total number of error records are %s" % (cached_log.count()))
print("Number of product pages visited that have Errors is %s" %
      (cached_log.filter(lambda x: "product" in x).count()))
```

Penjelasan: Kode program di atas merupakan program untuk melakukan analisis terhadap file log akses. Program ini menggunakan Apache Spark untuk membaca file log akses dan melakukan operasi pemrosesan data pada RDD

cloudera@quickstart:~/spark-2.0.0-bin-hadoop2.7

File Edit View Search Terminal Help

import sys.process._

scala> val output = "hadoop fs -ls" !!

warning: there was one feature warning; re-run with -feature for details

23/03/23 01:23:42 WARN ipc.Client: Failed to connect to server: quickstart.cloud

era/10.0.2.15:8020: try once and fail.

java.net.ConnectException: Connection refused

at sun.nio.ch.SocketChannelImpl.checkConnect(Native Method)

at sun.nio.ch.SocketChannelImpl.finishConnect(SocketChannelImpl.java:739

)

at org.apache.hadoop.net.SocketIOWithTimeout.connect(SocketIOWithTimeout

.java:206)

at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:530)

at org.apache.hadoop.net.NetUtils.connect(NetUtils.java:494)

at org.apache.hadoop.ipc.Client\$Connection.setupConnection(Client.java:6

48)

at org.apache.hadoop.ipc.Client\$Connection.setupIOstreams(Client.java:74

4)

at org.apache.hadoop.ipc.Client\$Connection.access\$3000(Client.java:396)

at org.apache.hadoop.ipc.Client.getConnection(Client.java:1557)

at org.apache.hadoop.ipc.Client.call(Client.java:1480)

at org.apache.hadoop.ipc.Client.call(Client.java:1441)

at org.apache.hadoop.ipc.ProtobufRpcEngine\$Invoker.invoke(ProtobufRpcEng

ine.java:230)

*Unsaved Document 1 - gedit

File Edit View Search Tools Documents Help

Open Save Undo

*Tugas Minggu 6 *Unsaved Document 1

Nama : Muhamad Al Kausar Ramadhan

Kelas : TI 3B

Nim : 2041720193

Percobaan SystemCommandsOutput

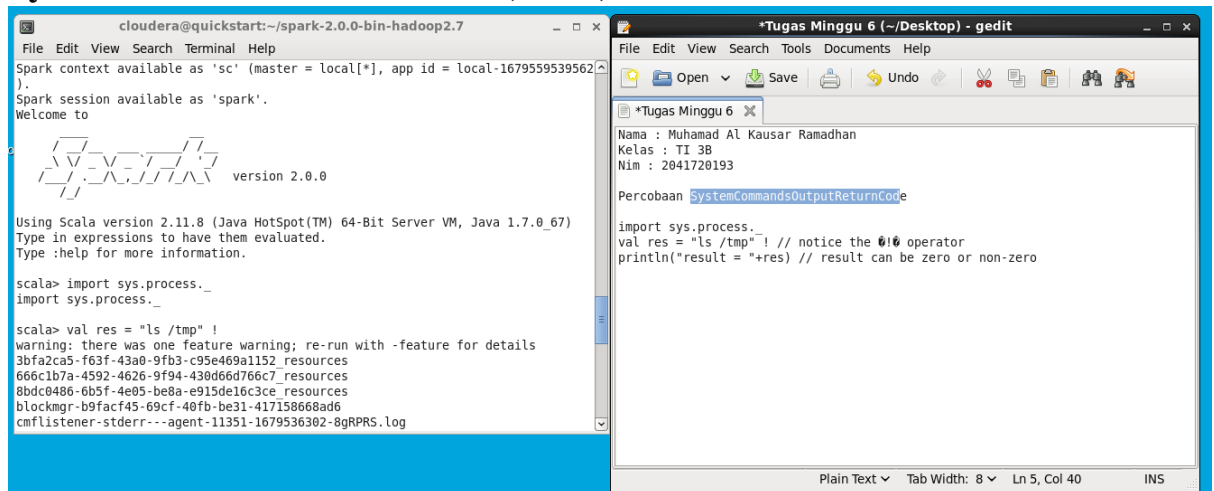
import sys.process._

val output = "hadoop fs -ls" !! // notice the !! operator

println("result = "+output)

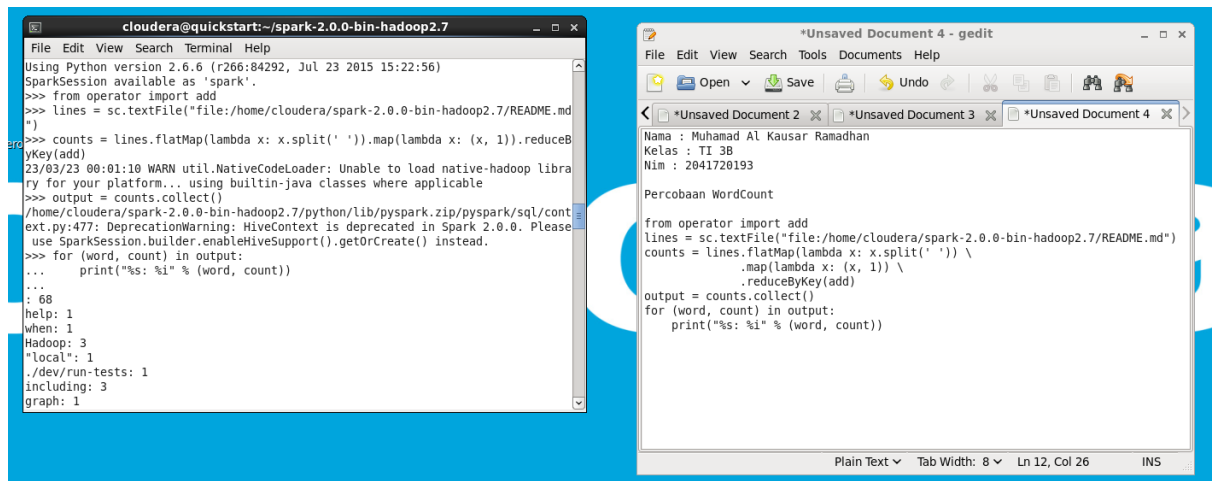
Plain Text Tab Width: 8 Ln 9, Col 28 INS

6. SystemCommandsReturnCode(Scala)



7. Understanding RDDs

8. WordCount



Penjelasan: Kode program di atas adalah contoh penggunaan Apache Spark menggunakan Python untuk menghitung jumlah kemunculan setiap kata pada sebuah file teks (dalam contoh ini menggunakan file README.md yang ada di folder home/cloudera/spark-2.0.0-bin-hadoop2.7). Kode program tersebut membaca file teks dari path tertentu, melakukan transformasi data dalam RDD (Resilient Distributed Dataset) dengan melakukan pemecahan string