BIG DATA WEEK 11 TASK REPORT
(JOBSHEET MAPREDUCE 3)

By:

Rajendra Rakha Arya Prabaswara

1941720080

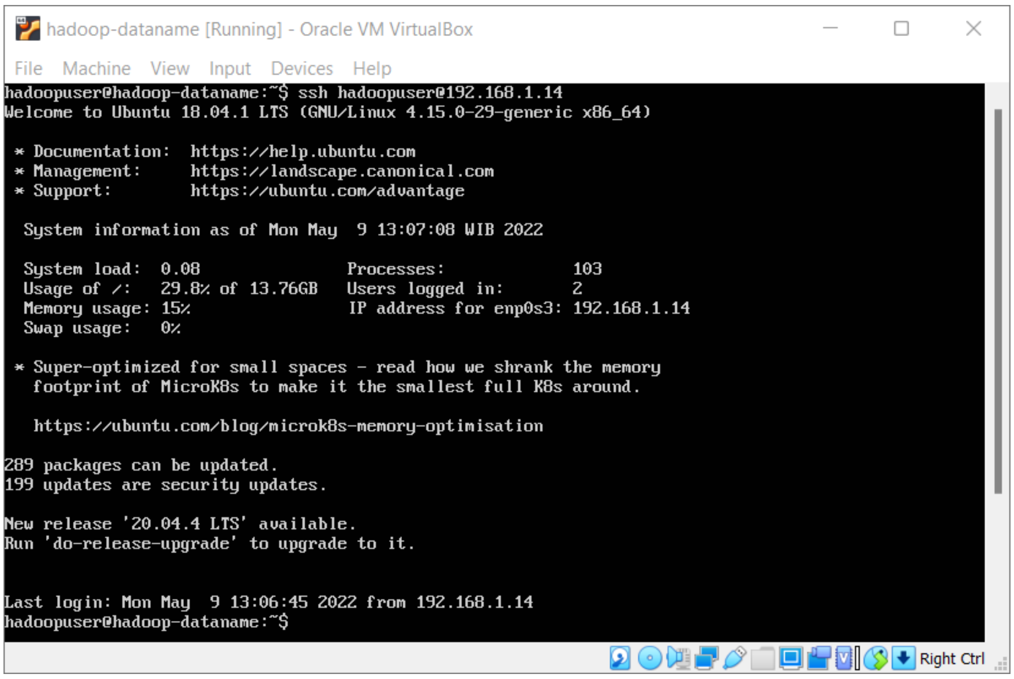
(TI-3H / 20)

INFORMATICS ENGINEERING STUDY PROGRAM MAJORING INFORMATION TECHNOLOGY

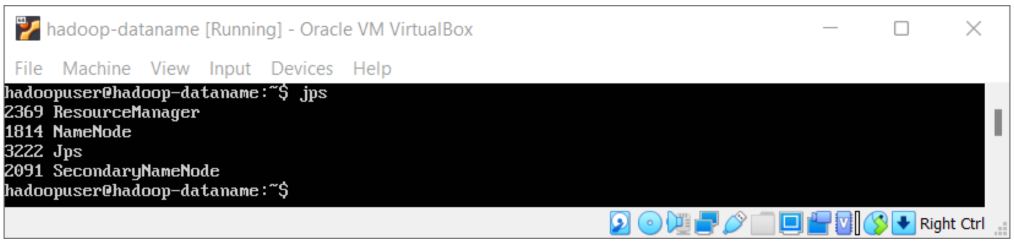
STATE POLYTECHNIC MALANG

Part I : Creating a MapReduce Job with JetBrains IntelliJ IDEA

1. Install and will be able to access the Hadoop cluster correctly .

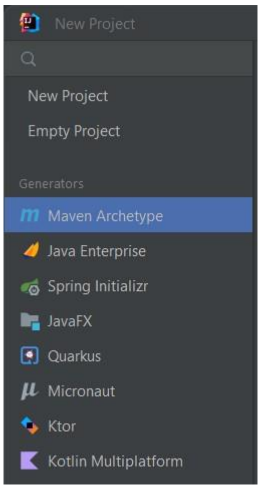

hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

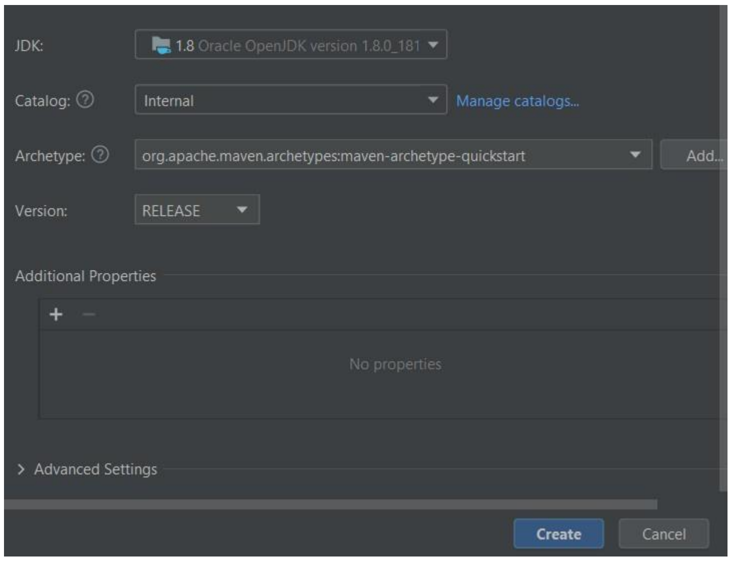
```
hadoopuser@hadoop-dataname:~$ ssh hadoopuser@192.168.1.14
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Mon May  9 13:07:08 WIB 2022

  System load:  0.08               Processes:             103
  Usage of /:   29.8% of 13.76GB   Users logged in:       2
  Memory usage: 15%                IP address for enp0s3: 192.168.1.14
  Swap usage:   0%

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

   https://ubuntu.com/blog/microk8s-memory-optimisation

289 packages can be updated.
199 updates are security updates.

New release '20.04.4 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon May  9 13:06:45 2022 from 192.168.1.14
hadoopuser@hadoop-dataname:~$
```


hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
hadoopuser@hadoop-dataname:~$ jps
2369 ResourceManager
1814 NameNode
3222 Jps
2091 SecondaryNameNode
hadoopuser@hadoop-dataname:~$
```

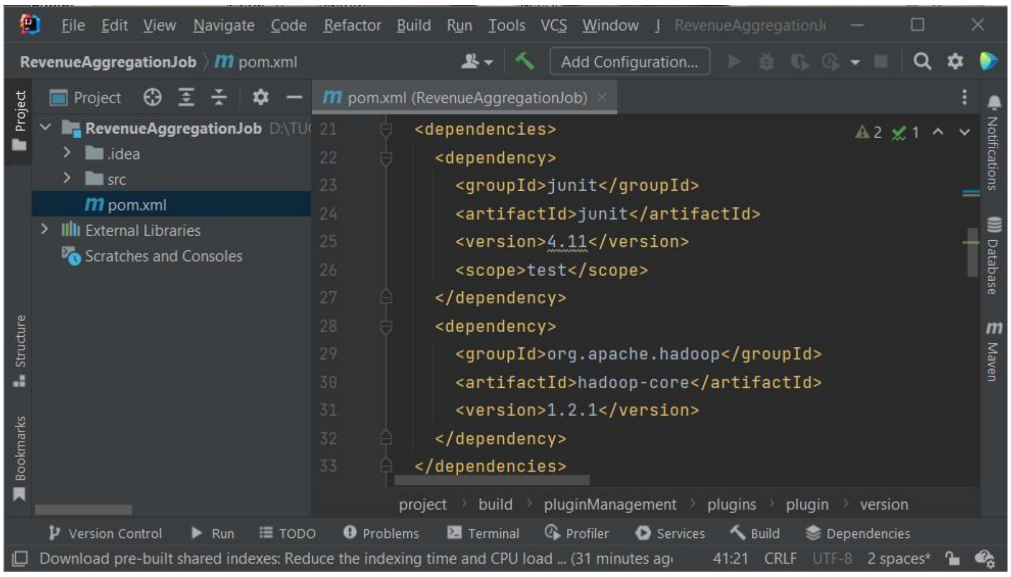
2. Open the IntelliJ IDEA IDE IDE and create a new project with the Java Language and maven framework build. And select the archetype "maven quick-start".





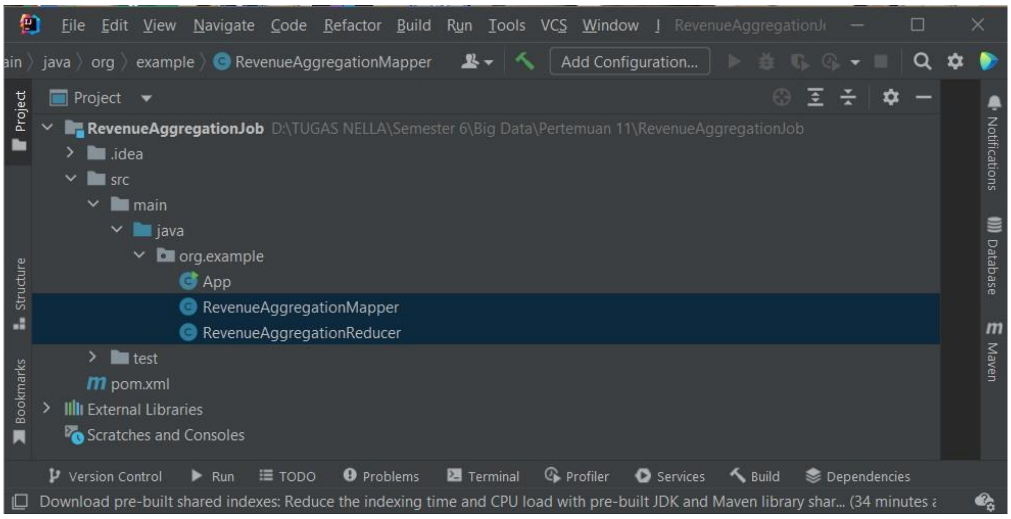
JDK: 1.8 Oracle OpenJDK version 1.8.0_181
Catalog: Internal
Archetype: org.apache.maven.archetypes:maven-archetype-quickstart
Version: RELEASE

3. After the project is completed. In pom.xml added dependencies in order to create a MapReduce Job. Reload the project by clicking on the 'm' icon.



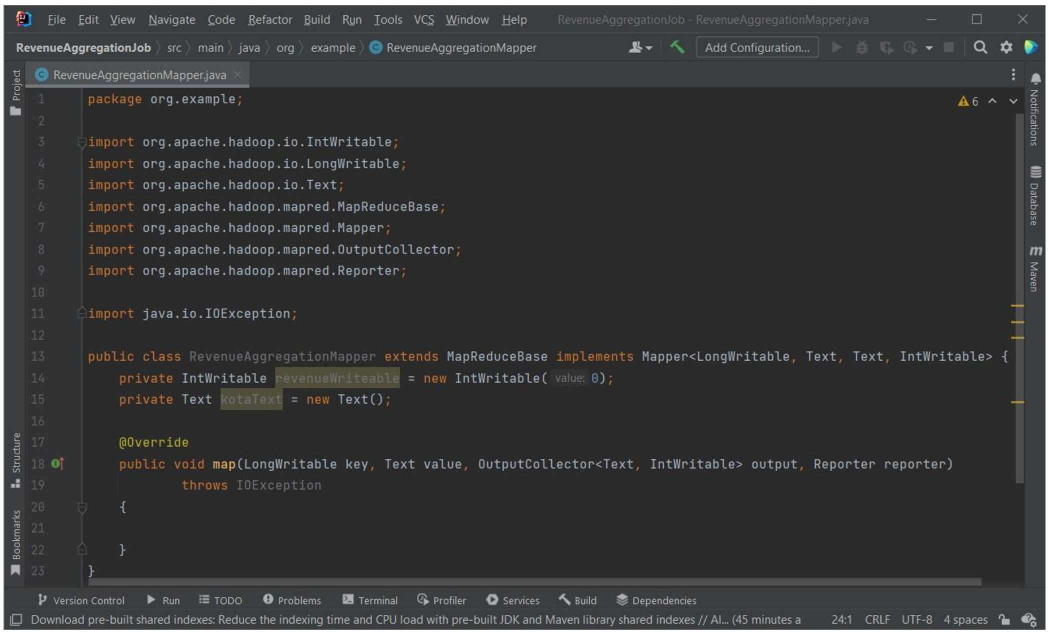
```xml
<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>4.11</version>
    <scope>test</scope>
  </dependency>
  <dependency>
    <groupId>org.apache.hadoop</groupId>
    <artifactId>hadoop-core</artifactId>
    <version>1.2.1</version>
  </dependency>
</dependencies>
```

4. Create 2 new classes on the project under the names RevenueAggregationMapper and RevenueAggregationReducer.



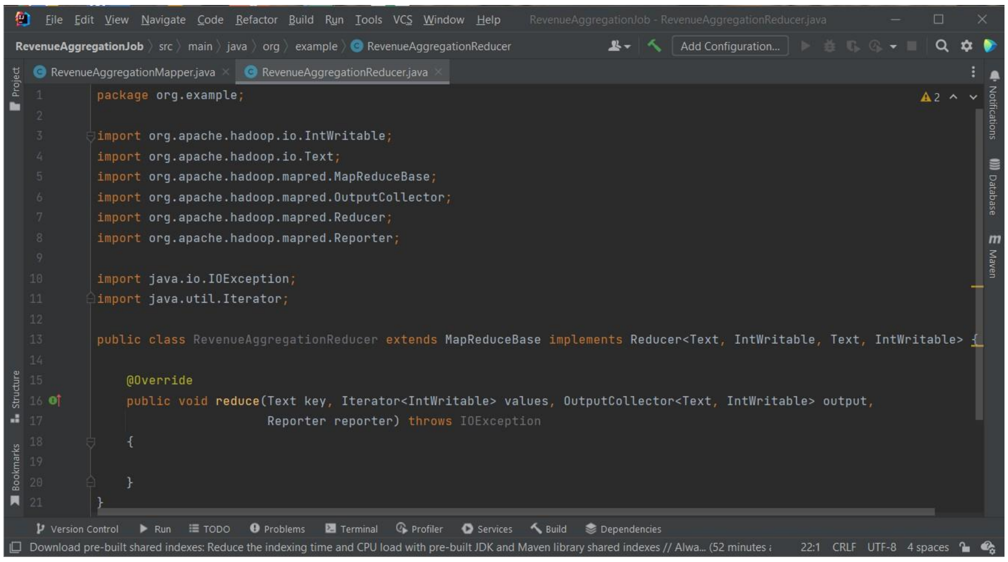
```
RevenueAggregationJob  D:\TUGAS NELLA\Semester 6\Big Data\Pertemuan 11\RevenueAggregationJob
  .idea
  src
    main
      java
        org.example
          App
          RevenueAggregationMapper
          RevenueAggregationReducer
    test
  pom.xml
External Libraries
Scratches and Consoles
```

5. Modify the Class RevenueAggregationMapper.



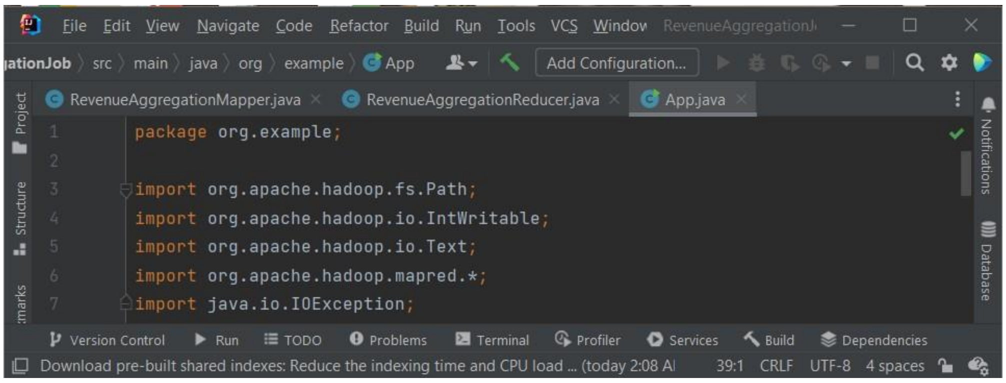
```java
package org.example;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;

import java.io.IOException;

public class RevenueAggregationMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
    private IntWritable revenueWriteable = new IntWritable( value: 0);
    private Text kotaText = new Text();

    @Override
    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter)
            throws IOException
    {

    }
}
```

6. Modify the Class RevenueAggregationReducer.



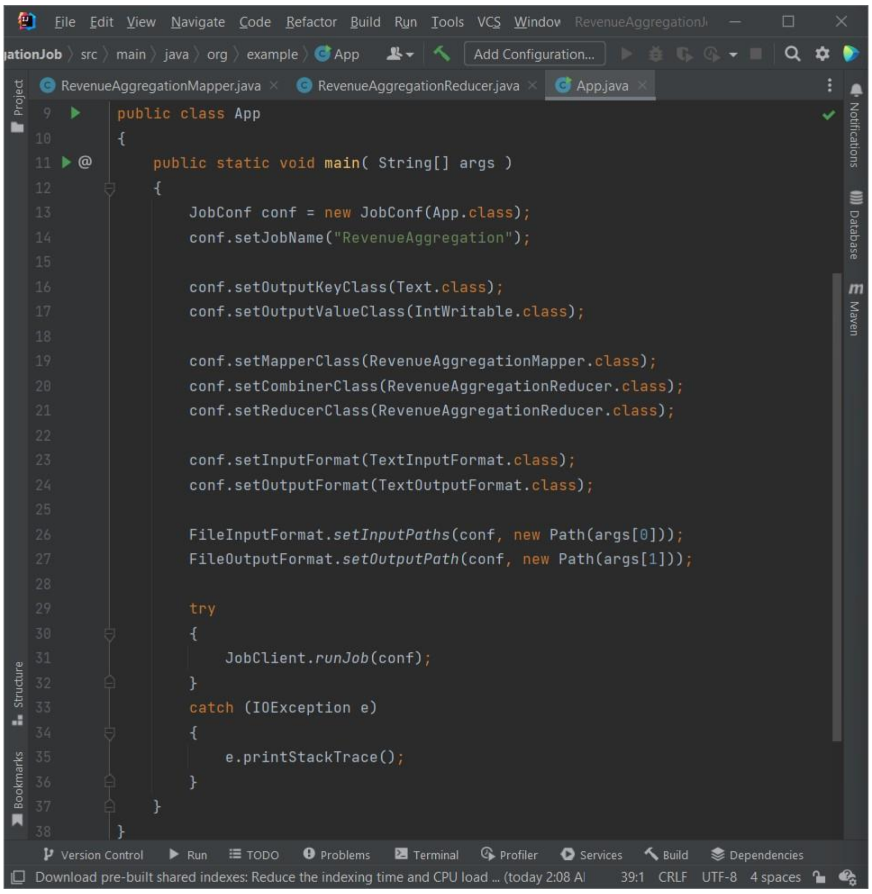
```java
package org.example;

import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;

import java.io.IOException;
import java.util.Iterator;

public class RevenueAggregationReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

    @Override
    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output,
                       Reporter reporter) throws IOException
    {

    }
}
```

7. Add imports blocks to the App class that is the project's built-in class.



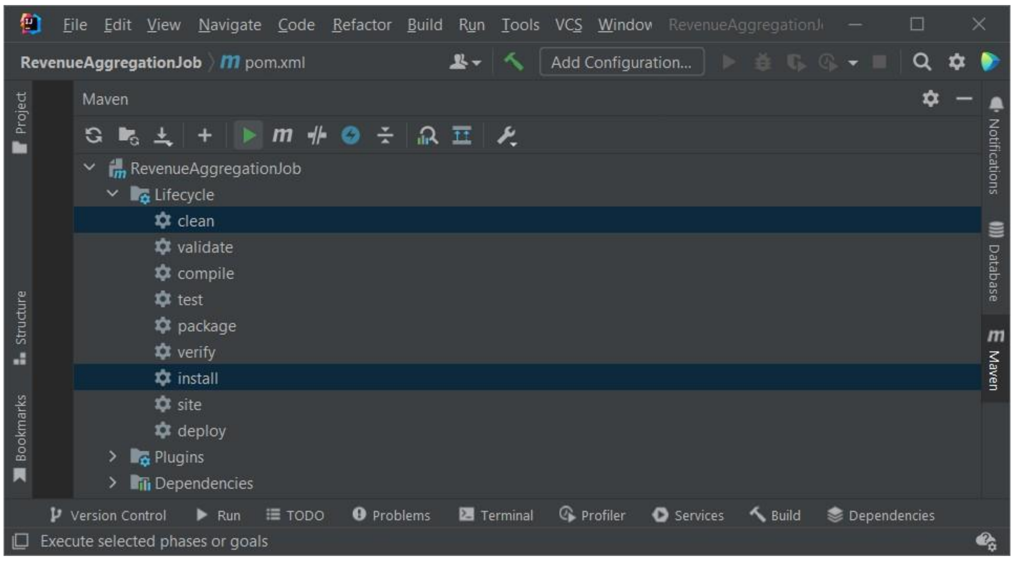
```java
package org.example;

import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.*;
import java.io.IOException;
```

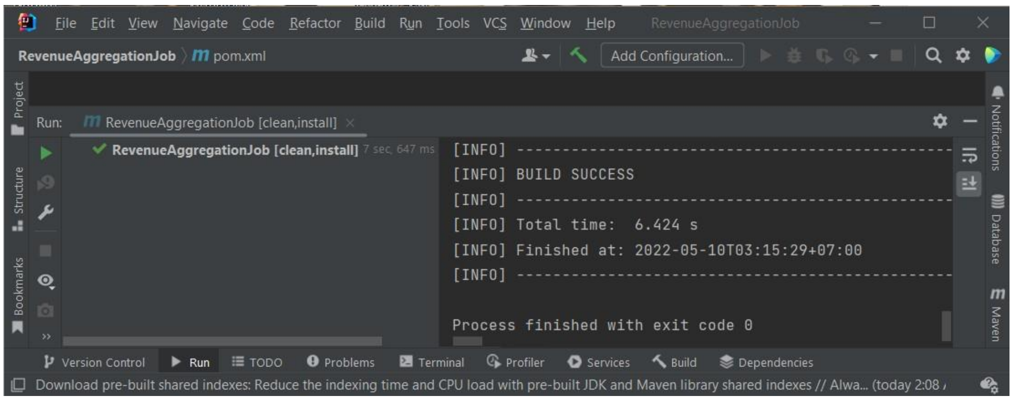
8. Then, add the code to the same class body .



```java
public class App
{
    public static void main( String[] args )
    {
        JobConf conf = new JobConf(App.class);
        conf.setJobName("RevenueAggregation");

        conf.setOutputKeyClass(Text.class);
        conf.setOutputValueClass(IntWritable.class);

        conf.setMapperClass(RevenueAggregationMapper.class);
        conf.setCombinerClass(RevenueAggregationReducer.class);
        conf.setReducerClass(RevenueAggregationReducer.class);

        conf.setInputFormat(TextInputFormat.class);
        conf.setOutputFormat(TextOutputFormat.class);

        FileInputFormat.setInputPaths(conf, new Path(args[0]));
        FileOutputFormat.setOutputPath(conf, new Path(args[1]));

        try
        {
            JobClient.runJob(conf);
        }
        catch (IOException e)
        {
            e.printStackTrace();
        }
    }
}
```

9. Compile the program using the help of Maven by opening the Maven panel located to the right of the IDE. In the pane expand the folder "Lifecycle", select "Clean" and "Installl". Then click Run.

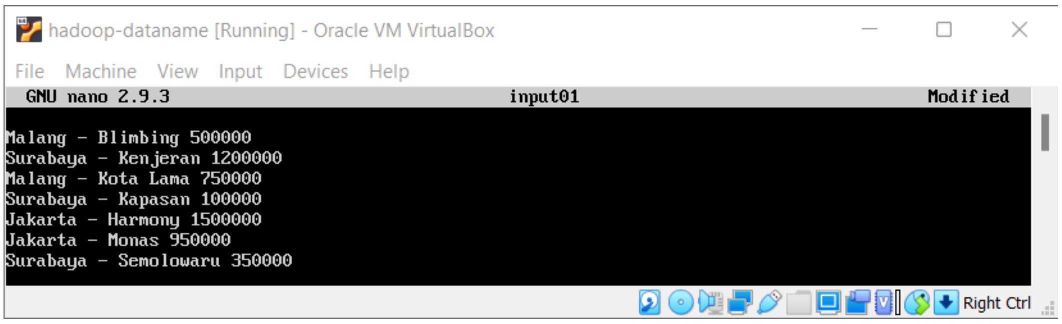



File Edit View Navigate Code Refactor Build Run Tools VCS Window Help RevenueAggregationJob
RevenueAggregationJob
pom.xml
Add Configuration...
Run: RevenueAggregationJob [clean,install]
RevenueAggregationJob [clean,install] 7 sec, 647 ms
Version Control
Run
TODO
Problems
Terminal
Profiler
Services
Build
Dependencies

```
[INFO] ------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------
[INFO] Total time:  6.424 s
[INFO] Finished at: 2022-05-10T03:15:29+07:00
[INFO] ------------------------------------------------------------

Process finished with exit code 0
```

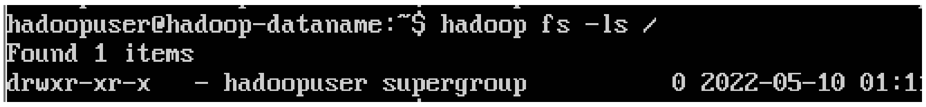
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Alwa... (today 2:08

10. Create a plain text file named input01.

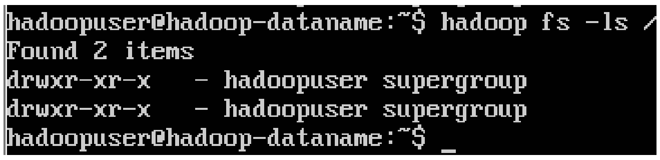

hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
  GNU nano 2.9.3                    input01                          Modified

Malang - Blimbing 500000
Surabaya - Kenjeran 1200000
Malang - Kota Lama 750000
Surabaya - Kapasan 100000
Jakarta - Harmony 1500000
Jakarta - Monas 950000
Surabaya - Semolowaru 350000
```

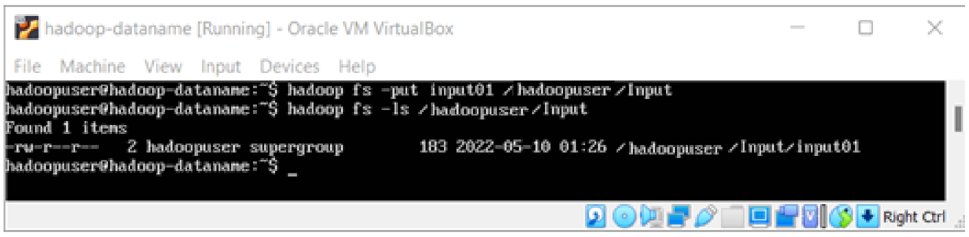
Upload the file to HDFS in the /YourName/Input folder. Also create 1 other folder in HDFS under the name /NamAnda/Output.



```
hadoopuser@hadoop-dataname:~$ hadoop fs -ls /
Found 1 items
drwxr-xr-x   - hadoopuser supergroup          0 2022-05-10 01:1
```

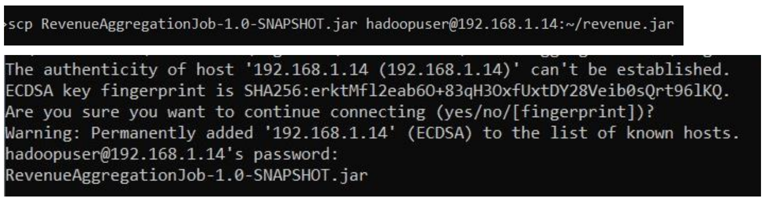


```
hadoopuser@hadoop-dataname:~$ hadoop fs -ls /
Found 2 items
drwxr-xr-x   - hadoopuser supergroup
drwxr-xr-x   - hadoopuser supergroup
hadoopuser@hadoop-dataname:~$ _
```


hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
hadoopuser@hadoop-dataname:~$ hadoop fs -put input01 /hadoopuser/Input
hadoopuser@hadoop-dataname:~$ hadoop fs -ls /hadoopuser/Input
Found 1 items
-rw-r--r--   2 hadoopuser supergroup        183 2022-05-10 01:26 /hadoopuser/Input/input01
hadoopuser@hadoop-dataname:~$ _
```

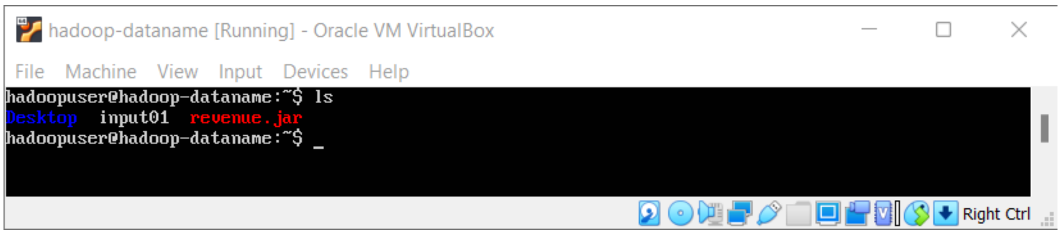
11. Upload jar files to the Name Node by using SCP or other methods. Execute the JAR using the command: Hadoop jar <nama_file>.jar <package_identifier> <folder_input_di_hdfs> <folder_output_di_hdfs>. JAR files uploaded to the namenode aredibe rinama revenue.jar.



```
scp RevenueAggregationJob-1.0-SNAPSHOT.jar hadoopuser@192.168.1.14:~/revenue.jar
```

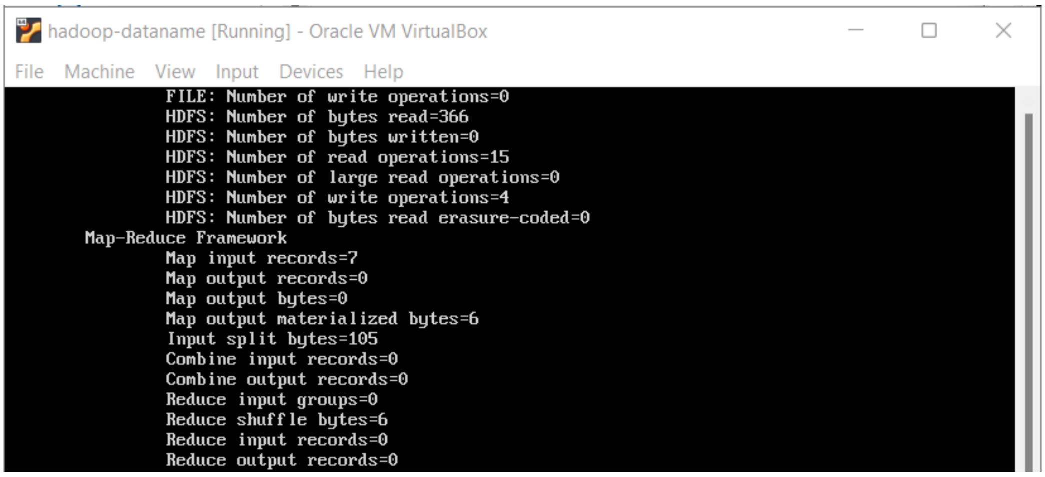
```
The authenticity of host '192.168.1.14 (192.168.1.14)' can't be established.
ECDSA key fingerprint is SHA256:erktMfl2eab6O+83qH3OxfUxtDY28Veib0sQrt96lKQ.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '192.168.1.14' (ECDSA) to the list of known hosts.
hadoopuser@192.168.1.14's password:
RevenueAggregationJob-1.0-SNAPSHOT.jar
```

12. When the execution process is complete, the folder /Your Name/Output/output/result in which there is a part-00000 file that when you

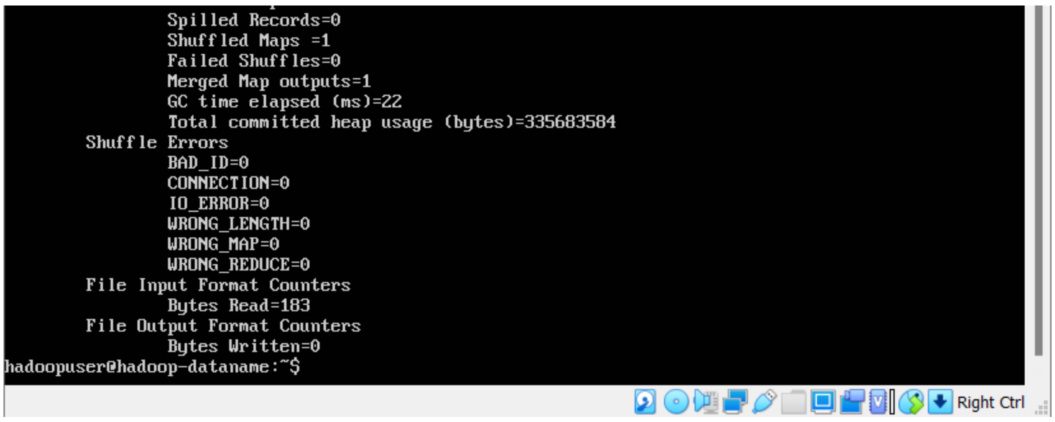

hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
hadoopuser@hadoop-dataname:~$ ls
Desktop  input01  revenue.jar
hadoopuser@hadoop-dataname:~$ _
```

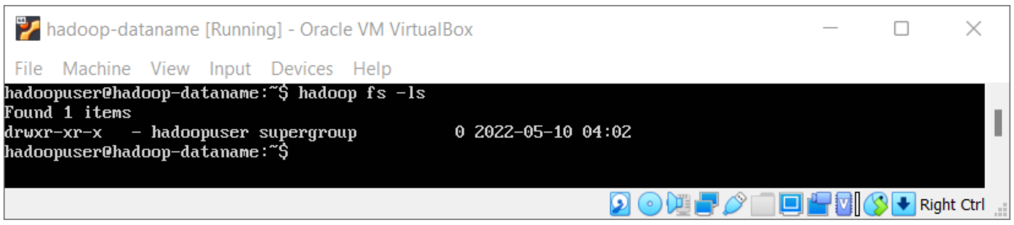
read, the contents are still empty. [Question] Why is the output file empty/has no content?


hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

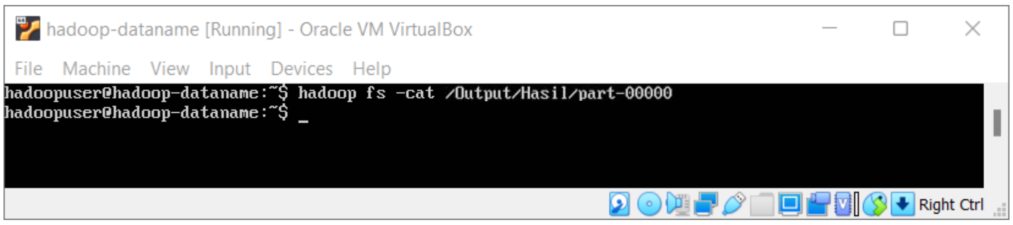
```
                FILE: Number of write operations=0
                HDFS: Number of bytes read=366
                HDFS: Number of bytes written=0
                HDFS: Number of read operations=15
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=4
                HDFS: Number of bytes read erasure-coded=0
        Map-Reduce Framework
                Map input records=7
                Map output records=0
                Map output bytes=0
                Map output materialized bytes=6
                Input split bytes=105
                Combine input records=0
                Combine output records=0
                Reduce input groups=0
                Reduce shuffle bytes=6
                Reduce input records=0
                Reduce output records=0
```



```
                Spilled Records=0
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=22
                Total committed heap usage (bytes)=335683584
        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=183
        File Output Format Counters
                Bytes Written=0
hadoopuser@hadoop-dataname:~$
```


hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

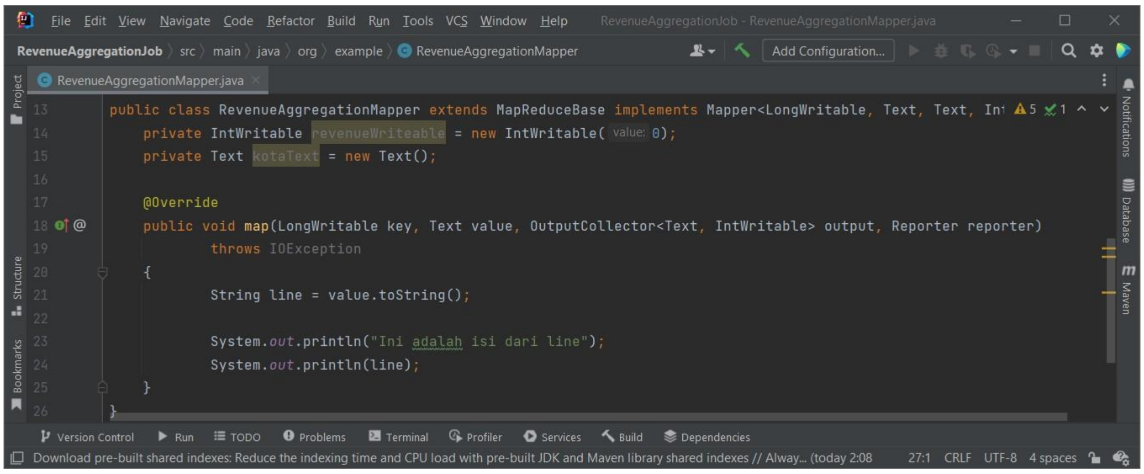
```
hadoopuser@hadoop-dataname:~$ hadoop fs -ls
Found 1 items
drwxr-xr-x   - hadoopuser supergroup          0 2022-05-10 04:02
hadoopuser@hadoop-dataname:~$
```


hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

```
hadoopuser@hadoop-dataname:~$ hadoop fs -cat /Output/Hasil/part-00000
hadoopuser@hadoop-dataname:~$ _
```

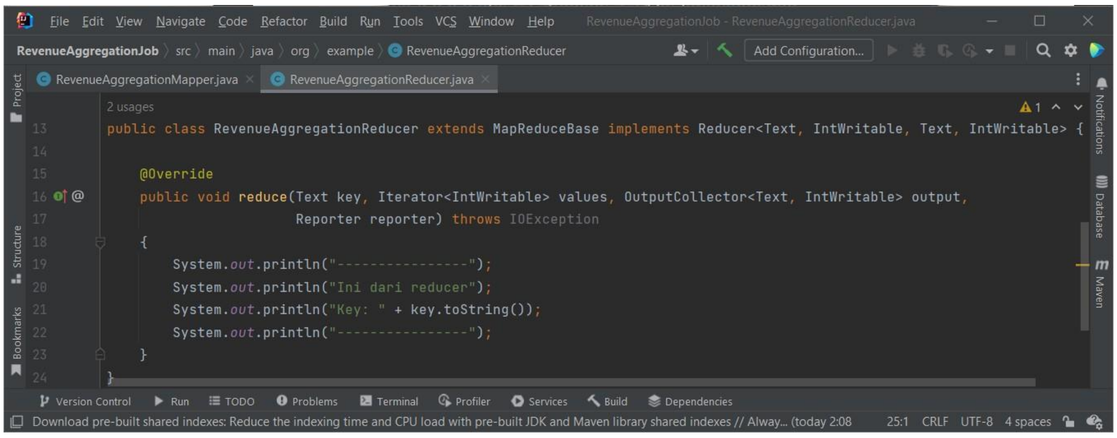
Bagian II : Mapper & Reducer

13. Modify the RevenueAggregationMapper class to find out how many times Mapper is run.


RevenueAggregationJob - RevenueAggregationMapper.java
RevenueAggregationJob src main java org example RevenueAggregationMapper
RevenueAggregationMapper.java

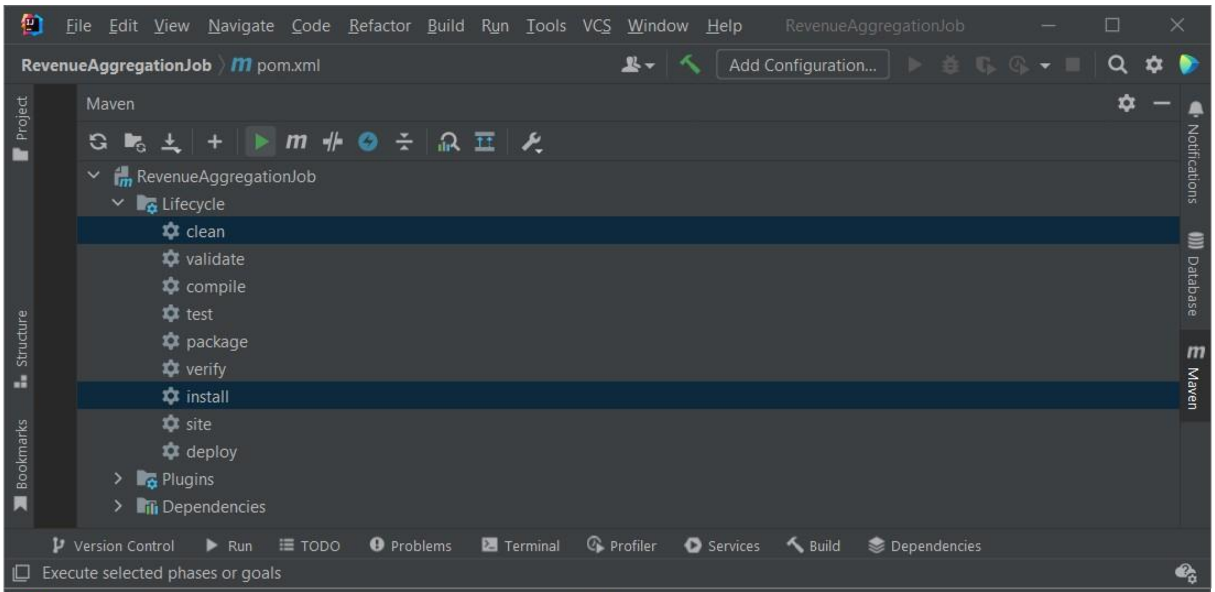
```java
public class RevenueAggregationMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, Int
    private IntWritable revenueWriteable = new IntWritable( value: 0);
    private Text kotaText = new Text();

    @Override
    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter)
            throws IOException
    {
            String line = value.toString();

            System.out.println("Ini adalah isi dari line");
            System.out.println(line);
    }
}
```

14. Modify the Class RevenueAggregationReducer.


RevenueAggregationJob - RevenueAggregationReducer.java
RevenueAggregationJob src main java org example RevenueAggregationReducer
RevenueAggregationMapper.java
RevenueAggregationReducer.java
2 usages

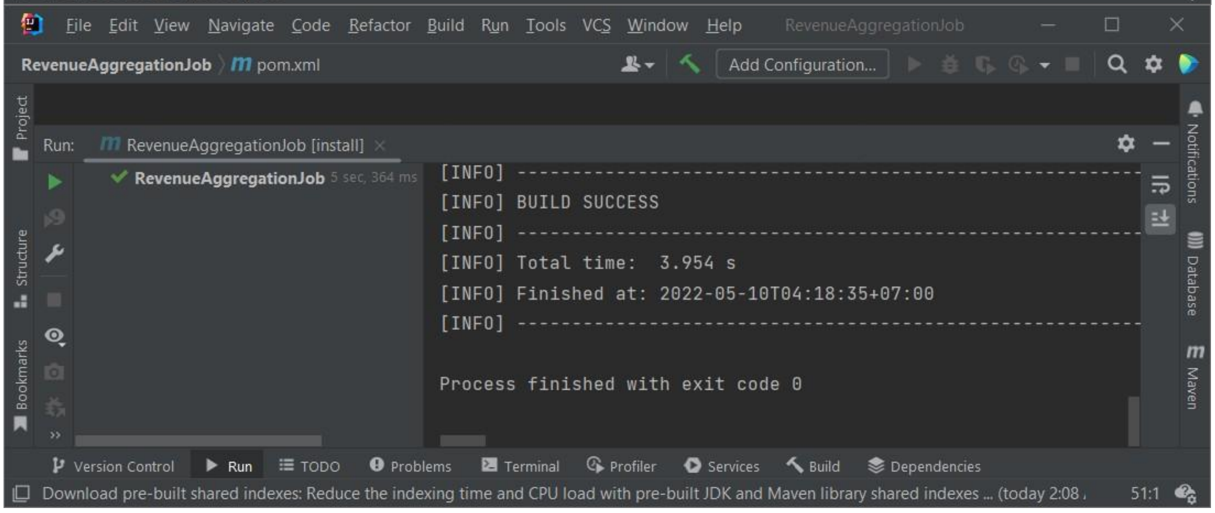
```java
public class RevenueAggregationReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable> {

    @Override
    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output,
                       Reporter reporter) throws IOException
    {
        System.out.println("---------------");
        System.out.println("Ini dari reducer");
        System.out.println("Key: " + key.toString());
        System.out.println("---------------");
    }
}
```

15. Rebuild jar again by using Maven.



```
RevenueAggregationJob
  Lifecycle
    clean
    validate
    compile
    test
    package
    verify
    install
    site
    deploy
  Plugins
  Dependencies
```

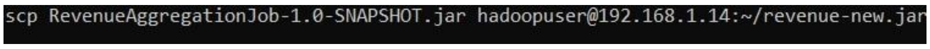
Execute selected phases or goals


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help RevenueAggregationJob
RevenueAggregationJob 〉 pom.xml
Add Configuration...
Run: RevenueAggregationJob [install]
RevenueAggregationJob 5 sec, 364 ms
Project
Structure
Bookmarks
Notifications
Database
Maven
Version Control
Run
TODO
Problems
Terminal
Profiler
Services
Build
Dependencies

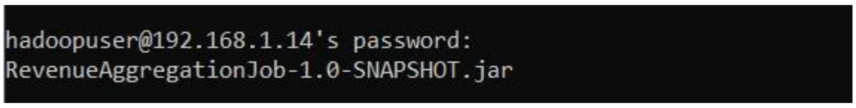
```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.954 s
[INFO] Finished at: 2022-05-10T04:18:35+07:00
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes ... (today 2:08 51:1

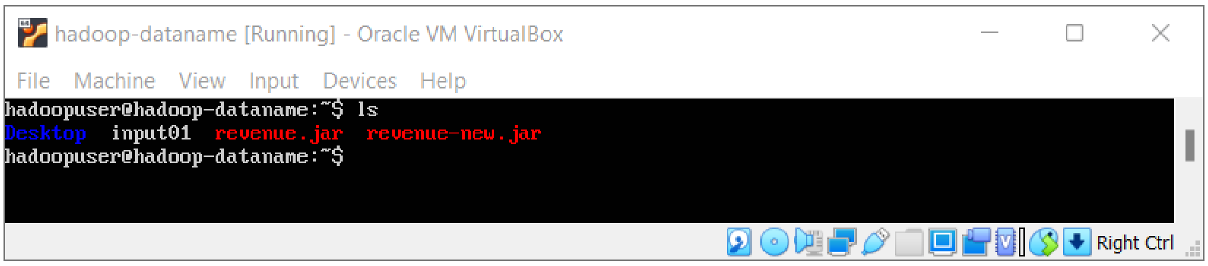
Re-upload the new JAR to the name node.



```
scp RevenueAggregationJob-1.0-SNAPSHOT.jar hadoopuser@192.168.1.14:~/revenue-new.jar
```

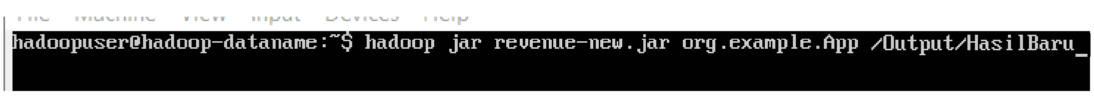


```
hadoopuser@192.168.1.14's password:
RevenueAggregationJob-1.0-SNAPSHOT.jar
```

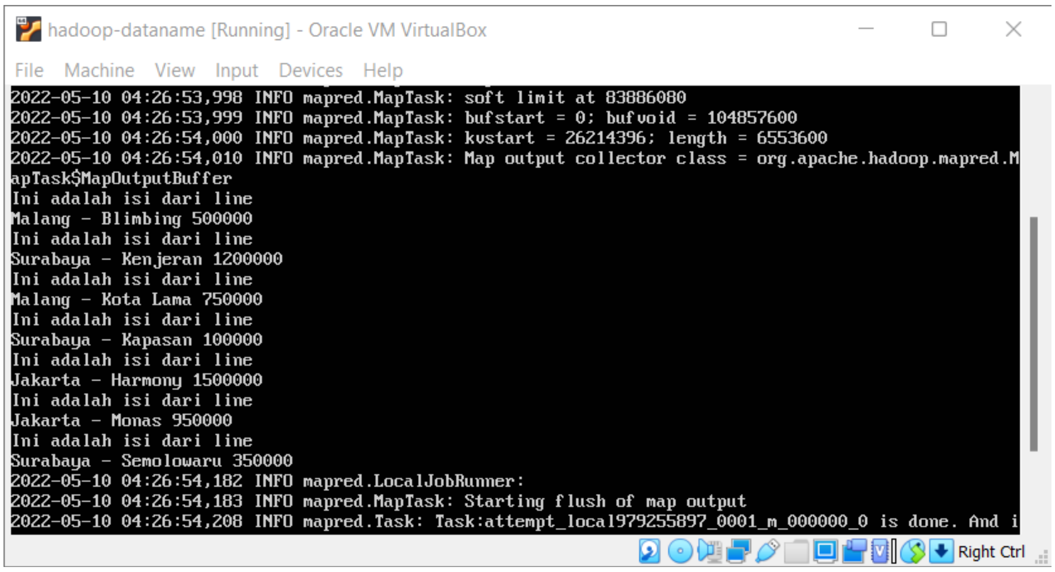

hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
hadoopuser@hadoop-dataname:~$ ls
Desktop  input01  revenue.jar  revenue-new.jar
hadoopuser@hadoop-dataname:~$
```

Execution.

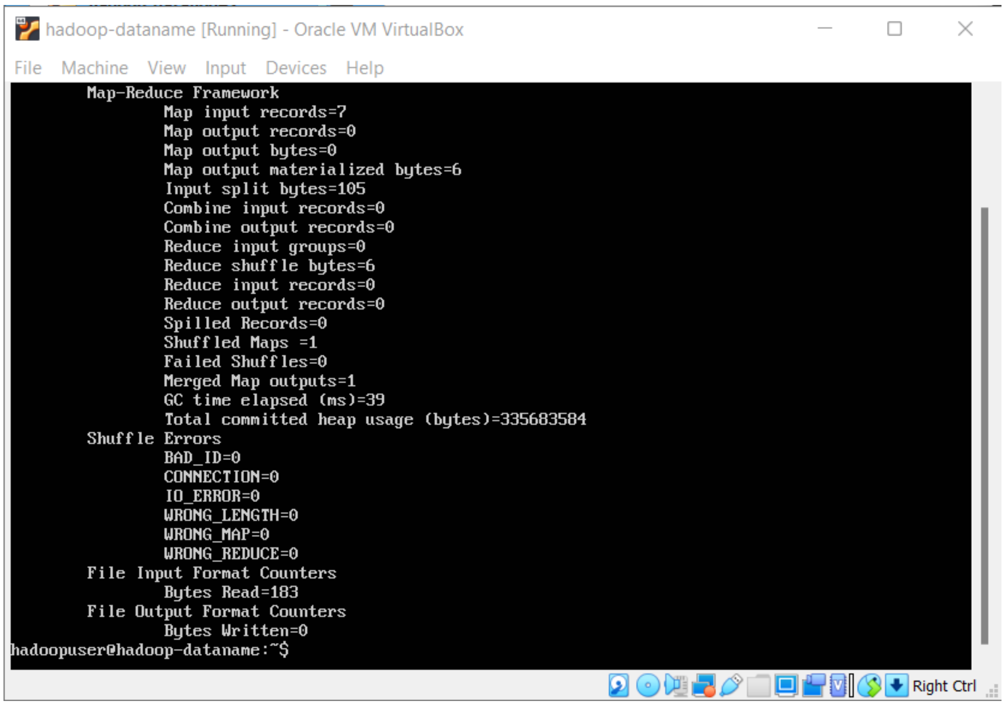


```
hadoopuser@hadoop-dataname:~$ hadoop jar revenue-new.jar org.example.App /Output/HasilBaru_
```


hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

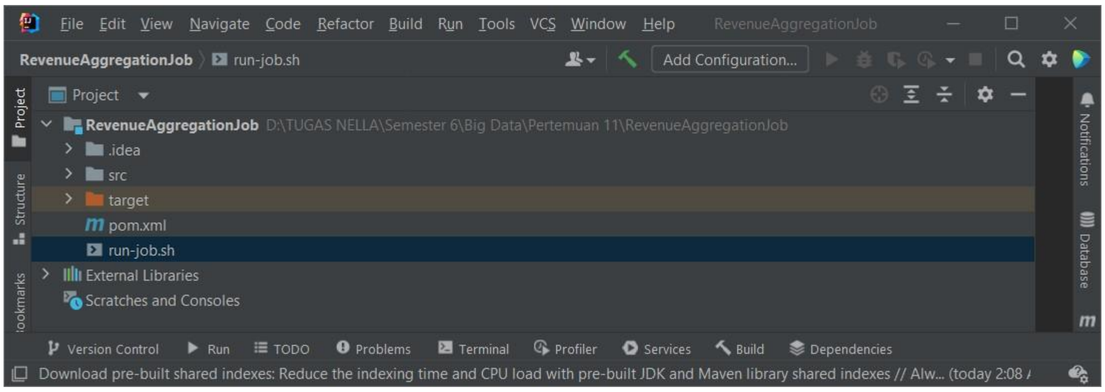
```
2022-05-10 04:26:53,998 INFO mapred.MapTask: soft limit at 83886080
2022-05-10 04:26:53,999 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2022-05-10 04:26:54,000 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2022-05-10 04:26:54,010 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.M
apTask$MapOutputBuffer
Ini adalah isi dari line
Malang - Blimbing 500000
Ini adalah isi dari line
Surabaya - Kenjeran 1200000
Ini adalah isi dari line
Malang - Kota Lama 750000
Ini adalah isi dari line
Surabaya - Kapasan 100000
Ini adalah isi dari line
Jakarta - Harmony 1500000
Ini adalah isi dari line
Jakarta - Monas 950000
Ini adalah isi dari line
Surabaya - Semolowaru 350000
2022-05-10 04:26:54,182 INFO mapred.LocalJobRunner:
2022-05-10 04:26:54,183 INFO mapred.MapTask: Starting flush of map output
2022-05-10 04:26:54,208 INFO mapred.Task: Task:attempt_local979255897_0001_m_000000_0 is done. And i
```

16. Scroll and/or search (Ctrl + F) on your console, is there a bookmark message from the Reducer class?

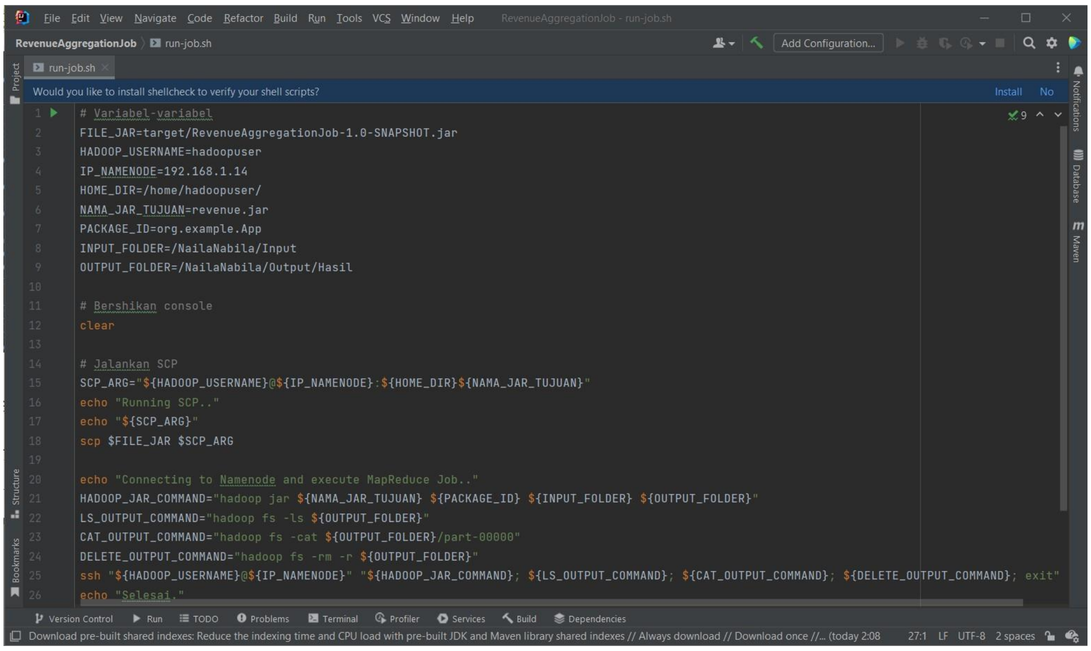

hadoop-dataname [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Right Ctrl

```
        Map-Reduce Framework
                Map input records=7
                Map output records=0
                Map output bytes=0
                Map output materialized bytes=6
                Input split bytes=105
                Combine input records=0
                Combine output records=0
                Reduce input groups=0
                Reduce shuffle bytes=6
                Reduce input records=0
                Reduce output records=0
                Spilled Records=0
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=39
                Total committed heap usage (bytes)=335683584
        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=183
        File Output Format Counters
                Bytes Written=0
hadoopuser@hadoop-dataname:~$
```

17. Add a new file. The name of the file: "run-job.sh"


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help RevenueAggregationJob
RevenueAggregationJob 〉 run-job.sh
Add Configuration...
Project
RevenueAggregationJob D:\TUGAS NELLA\Semester 6\Big Data\Pertemuan 11\RevenueAggregationJob
.idea
src
target
pom.xml
run-job.sh
External Libraries
Scratches and Consoles
Project
Structure
Bookmarks
Notifications
Database
Version Control
Run
TODO
Problems
Terminal
Profiler
Services
Build
Dependencies
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Alw... (today 2:08 /

18. Add a script to the file. Adjust the variables in the initial rows of the file to the conditions in the computer and cluster.

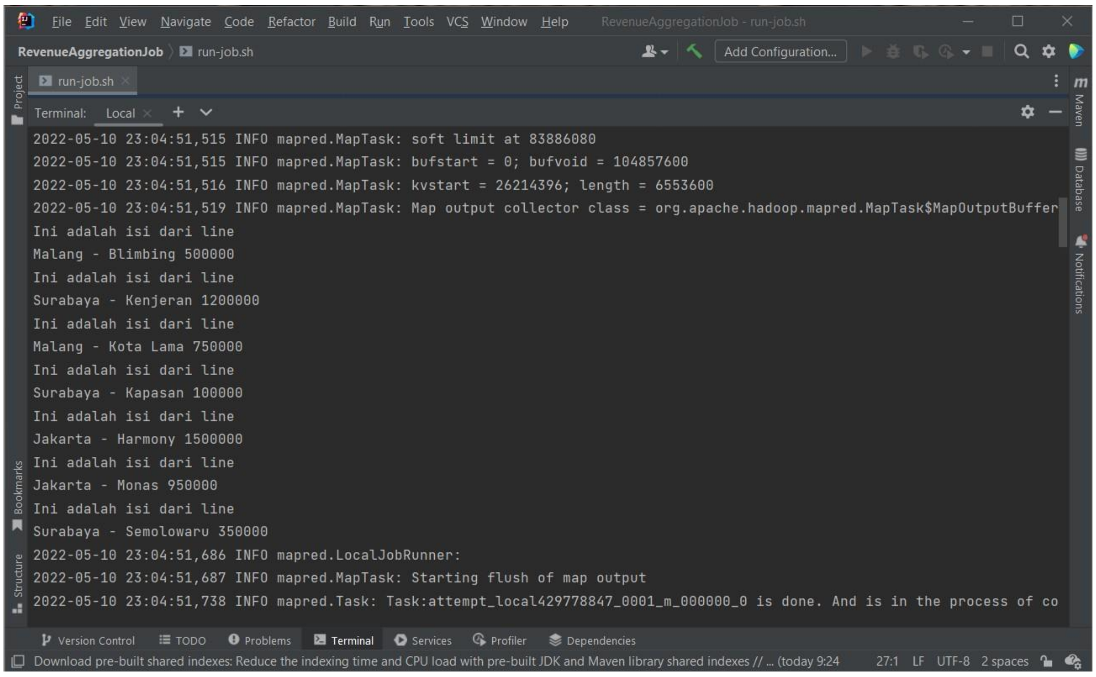


Would you like to install shellcheck to verify your shell scripts?

```
# Variabel-variabel
FILE_JAR=target/RevenueAggregationJob-1.0-SNAPSHOT.jar
HADOOP_USERNAME=hadoopuser
IP_NAMENODE=192.168.1.14
HOME_DIR=/home/hadoopuser/
NAMA_JAR_TUJUAN=revenue.jar
PACKAGE_ID=org.example.App
INPUT_FOLDER=/NailaNabila/Input
OUTPUT_FOLDER=/NailaNabila/Output/Hasil

# Bershikan console
clear

# Jalankan SCP
SCP_ARG="${HADOOP_USERNAME}@${IP_NAMENODE}:${HOME_DIR}${NAMA_JAR_TUJUAN}"
echo "Running SCP.."
echo "${SCP_ARG}"
scp $FILE_JAR $SCP_ARG

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

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Always download // Download once //... (today 2:08

19. Execution

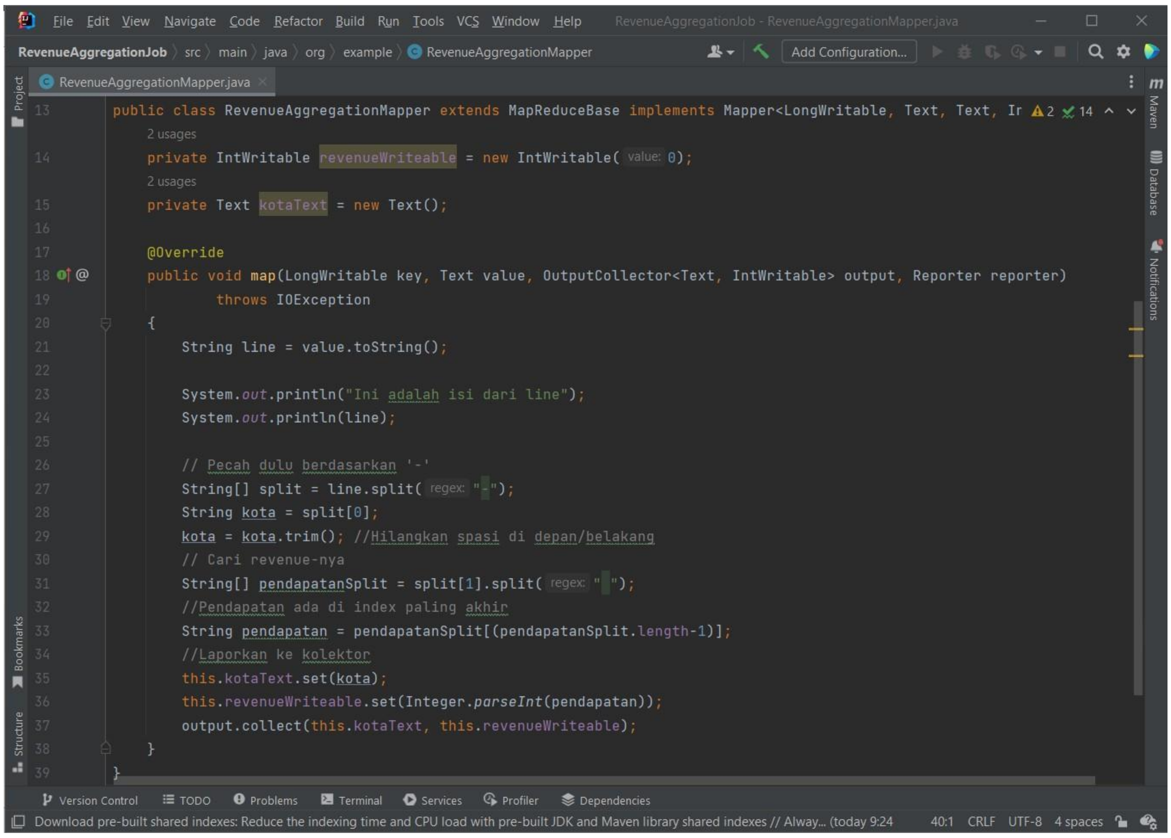


```
2022-05-10 23:04:51,515 INFO mapred.MapTask: soft limit at 83886080
2022-05-10 23:04:51,515 INFO mapred.MapTask: bufstart = 0; bufvoid = 104857600
2022-05-10 23:04:51,516 INFO mapred.MapTask: kvstart = 26214396; length = 6553600
2022-05-10 23:04:51,519 INFO mapred.MapTask: Map output collector class = org.apache.hadoop.mapred.MapTask$MapOutputBuffer
Ini adalah isi dari line
Malang - Blimbing 500000
Ini adalah isi dari line
Surabaya - Kenjeran 1200000
Ini adalah isi dari line
Malang - Kota Lama 750000
Ini adalah isi dari line
Surabaya - Kapasan 100000
Ini adalah isi dari line
Jakarta - Harmony 1500000
Ini adalah isi dari line
Jakarta - Monas 950000
Ini adalah isi dari line
Surabaya - Semolowaru 350000
2022-05-10 23:04:51,686 INFO mapred.LocalJobRunner:
2022-05-10 23:04:51,687 INFO mapred.MapTask: Starting flush of map output
2022-05-10 23:04:51,738 INFO mapred.Task: Task:attempt_local429778847_0001_m_000000_0 is done. And is in the process of co
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // ... (today 9:24

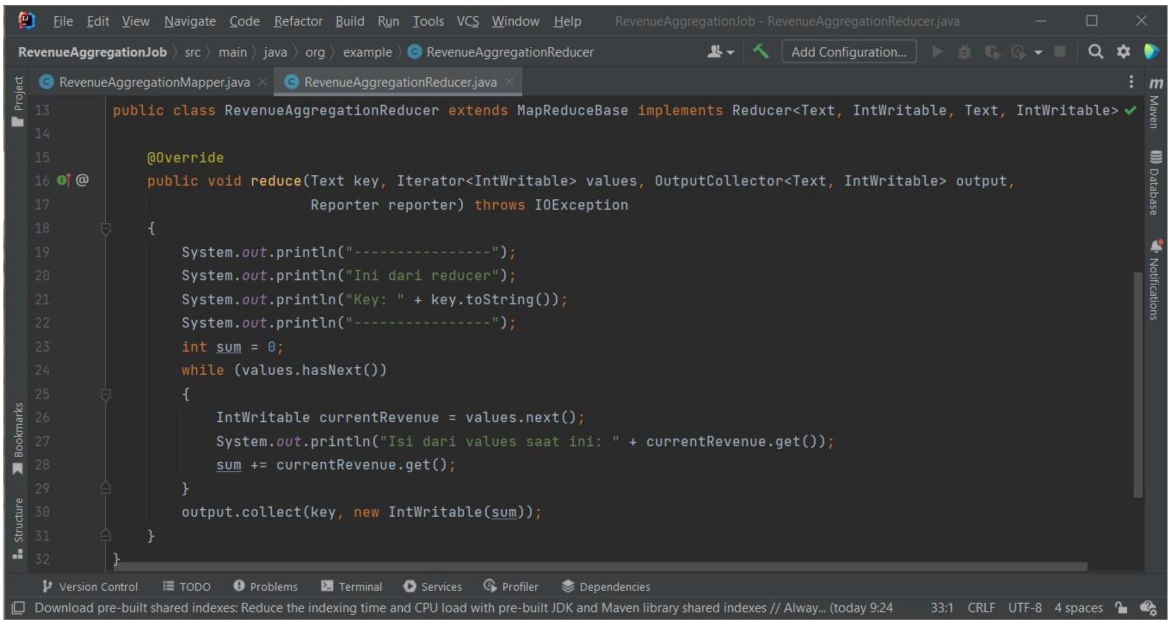
Part III: Logika Aggregation Process

20. Modify the Class RevenueAggregationMapper.



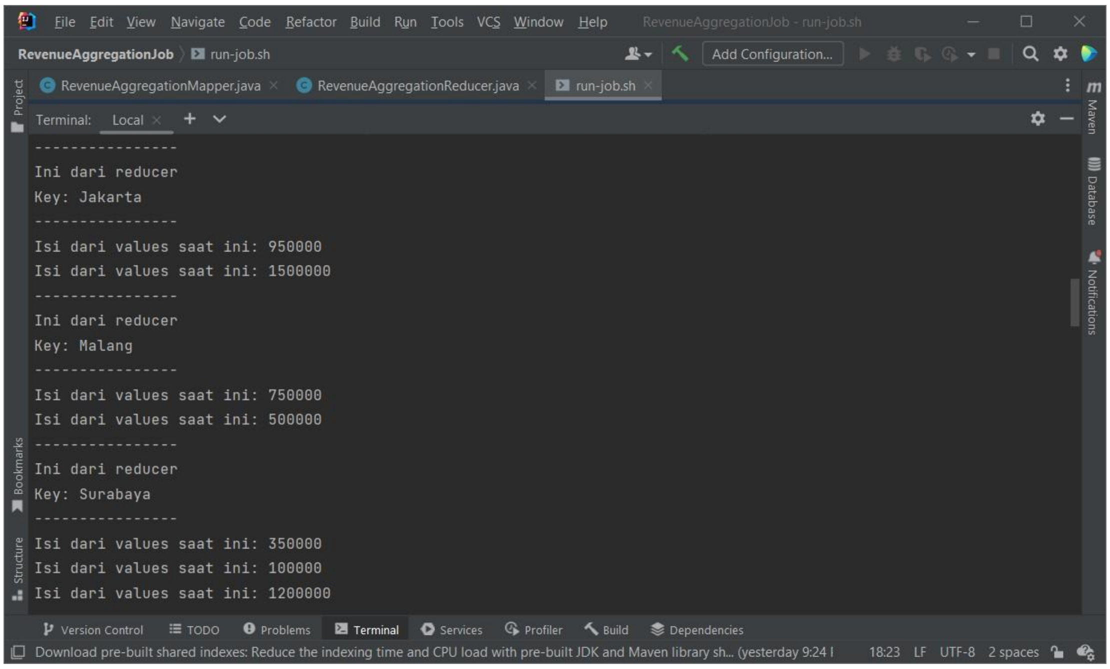
```java
public class RevenueAggregationMapper extends MapReduceBase implements Mapper<LongWritable, Text, Text, Ir
    private IntWritable revenueWriteable = new IntWritable(0);
    private Text kotaText = new Text();

    @Override
    public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output, Reporter reporter)
            throws IOException
    {
        String line = value.toString();

        System.out.println("Ini adalah isi dari line");
        System.out.println(line);

        // Pecah dulu berdasarkan '-'
        String[] split = line.split("-");
        String kota = split[0];
        kota = kota.trim(); //Hilangkan spasi di depan/belakang
        // Cari revenue-nya
        String[] pendapatanSplit = split[1].split(" ");
        //Pendapatan ada di index paling akhir
        String pendapatan = pendapatanSplit[(pendapatanSplit.length-1)];
        //Laporkan ke kolektor
        this.kotaText.set(kota);
        this.revenueWriteable.set(Integer.parseInt(pendapatan));
        output.collect(this.kotaText, this.revenueWriteable);
    }
}
```

21. In the RevenueAggregationReducer class, modify the reduce() method.



```java
public class RevenueAggregationReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable>

    @Override
    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output,
                       Reporter reporter) throws IOException
    {
        System.out.println("----------------");
        System.out.println("Ini dari reducer");
        System.out.println("Key: " + key.toString());
        System.out.println("----------------");
        int sum = 0;
        while (values.hasNext())
        {
            IntWritable currentRevenue = values.next();
            System.out.println("Isi dari values saat ini: " + currentRevenue.get());
            sum += currentRevenue.get();
        }
        output.collect(key, new IntWritable(sum));
    }
}
```

22. Compile, deploy and rerun the MapReduce program and pay attention to the results. Print marker messages from the Reducer class can find them in the console. Which means, the reduce() method in the class, this time called.

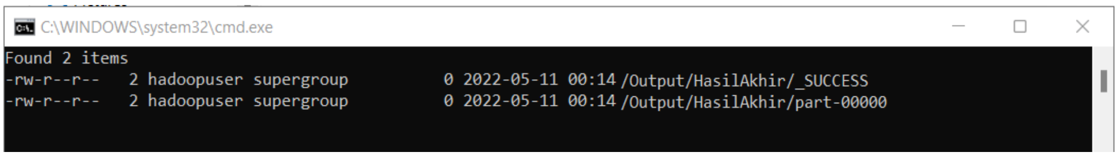


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
RevenueAggregationJob - run-job.sh
RevenueAggregationJob run-job.sh
Add Configuration...
RevenueAggregationMapper.java
RevenueAggregationReducer.java
run-job.sh
Terminal: Local
Version Control
TODO
Problems
Terminal
Services
Profiler
Build
Dependencies

```
---------------
Ini dari reducer
Key: Jakarta
---------------
Isi dari values saat ini: 950000
Isi dari values saat ini: 1500000
---------------
Ini dari reducer
Key: Malang
---------------
Isi dari values saat ini: 750000
Isi dari values saat ini: 500000
---------------
Ini dari reducer
Key: Surabaya
---------------
Isi dari values saat ini: 350000
Isi dari values saat ini: 100000
Isi dari values saat ini: 1200000
```

```
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library sh... (yesterday 9:24 l    18:23  LF  UTF-8  2 spaces
```

23. Scroll to thebottom of the console, now the results of aggregation should be clearly visible and correct results.



```
Found 2 items
-rw-r--r--   2 hadoopuser supergroup          0 2022-05-11 00:14 /Output/HasilAkhir/_SUCCESS
-rw-r--r--   2 hadoopuser supergroup          0 2022-05-11 00:14 /Output/HasilAkhir/part-00000
```

Part I Questions

1. Why is the output file empty/has no content?

   Answer:

   - **Because in the Mapper class and reducer class do not havea command to display the data to be stored, so the process is only run without being stored in the Results directory .**

Question Part II

1. When and how many folders() in the Mapper class are executed?

   Answer:

   - **Method map() is executed when it can be input and the method will be executed as much data from the input.**

2. Why is the reduce() method in the Reducer class never followed?

   Answer:

   - **Because the mapper result data is not sent into the Reducer class so the reduce() method does not have the data to be executed.**

Question Part III

1. When and how much is the reduce() method run?

   Answer:

   - **The method will be run when it can be input from the Mapper class and run 3 times because there are 3 keys, namely Malang, Surabaya, and Jakarta.**

2. Why can the console appear the contents of part-0000 files?

   Answer:

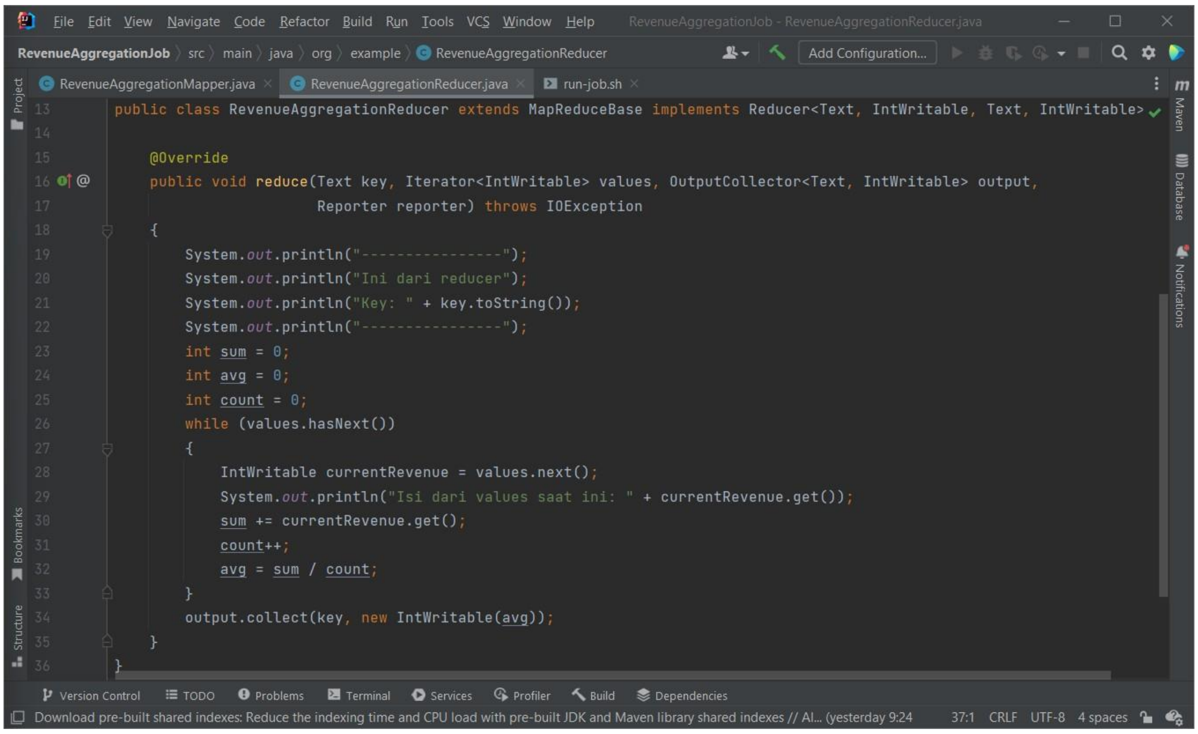
   - **Because the reducer result was successfully stored on the part-0000 file.**

Assignment

Try to replace its aggregation operations to be average per city instead of counting the totals!

Task Result:

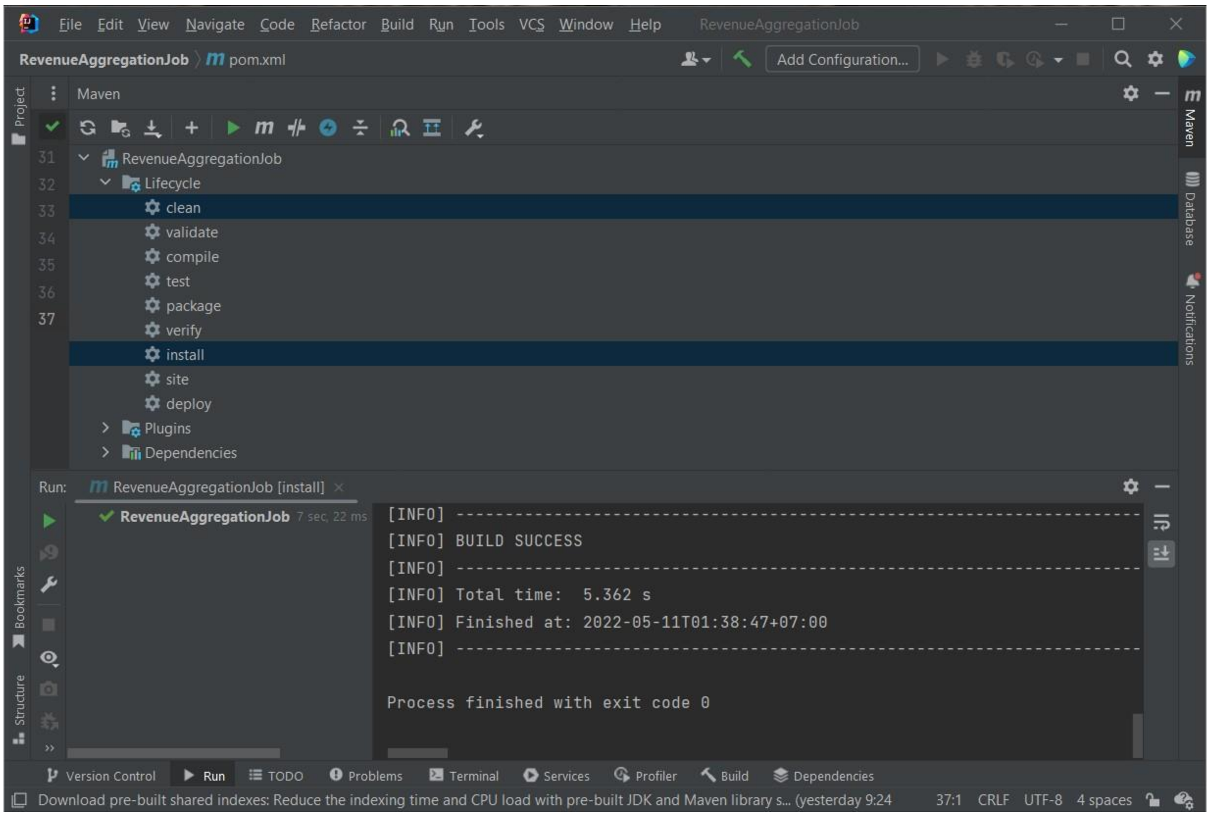
- Step 1: Modification of the RevenueAggregationReducer class


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
RevenueAggregationJob - RevenueAggregationReducer.java
RevenueAggregationJob › src › main › java › org › example › RevenueAggregationReducer
Add Configuration...
RevenueAggregationMapper.java
RevenueAggregationReducer.java
run-job.sh
Version Control
TODO
Problems
Terminal
Services
Profiler
Build
Dependencies

```java
public class RevenueAggregationReducer extends MapReduceBase implements Reducer<Text, IntWritable, Text, IntWritable>

    @Override
    public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, IntWritable> output,
                       Reporter reporter) throws IOException
    {
        System.out.println("---------------");
        System.out.println("Ini dari reducer");
        System.out.println("Key: " + key.toString());
        System.out.println("---------------");
        int sum = 0;
        int avg = 0;
        int count = 0;
        while (values.hasNext())
        {
            IntWritable currentRevenue = values.next();
            System.out.println("Isi dari values saat ini: " + currentRevenue.get());
            sum += currentRevenue.get();
            count++;
            avg = sum / count;
        }
        output.collect(key, new IntWritable(avg));
    }
}
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library shared indexes // Al... (yesterday 9:24   37:1 CRLF UTF-8 4 spaces

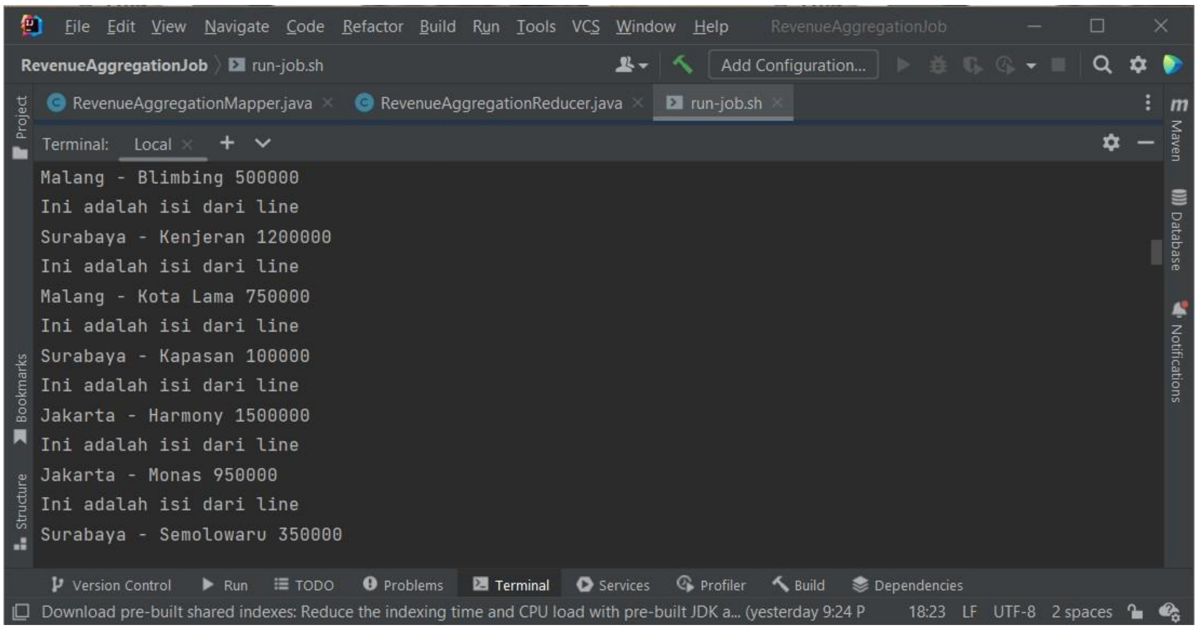
- Step 2: Compile and Build Project


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help RevenueAggregationJob
RevenueAggregationJob pom.xml
Add Configuration...
Maven
RevenueAggregationJob
Lifecycle
clean
validate
compile
test
package
verify
install
site
deploy
Plugins
Dependencies
Run: RevenueAggregationJob [install]
RevenueAggregationJob 7 sec, 22 ms
Version Control Run TODO Problems Terminal Services Profiler Build Dependencies

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  5.362 s
[INFO] Finished at: 2022-05-11T01:38:47+07:00
[INFO] ------------------------------------------------------------------------

Process finished with exit code 0
```

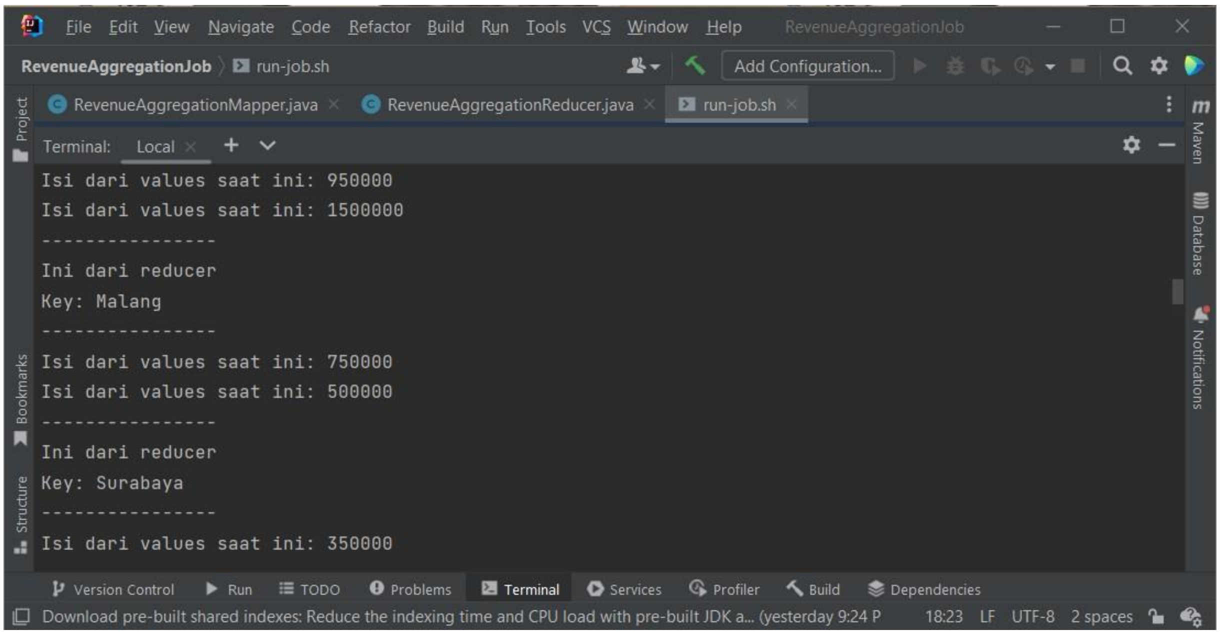
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK and Maven library s... (yesterday 9:24 37:1 CRLF UTF-8 4 spaces

- Startl


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help RevenueAggregationJob
RevenueAggregationJob run-job.sh
Add Configuration...
RevenueAggregationMapper.java
RevenueAggregationReducer.java
run-job.sh
Terminal: Local
Version Control Run TODO Problems Terminal Services Profiler Build Dependencies

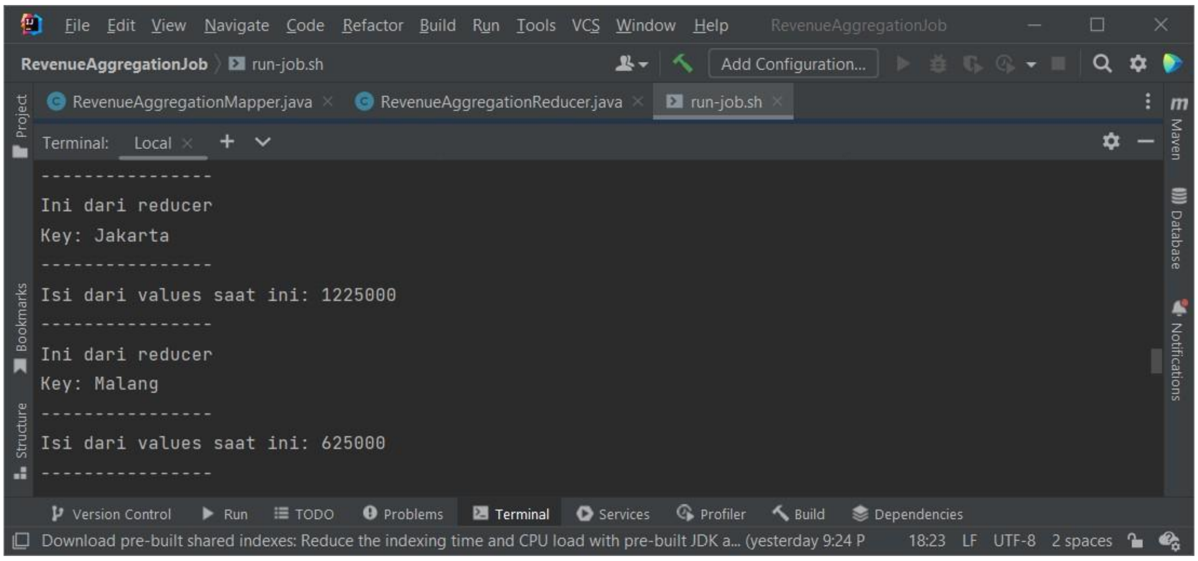
```
Malang - Blimbing 500000
Ini adalah isi dari line
Surabaya - Kenjeran 1200000
Ini adalah isi dari line
Malang - Kota Lama 750000
Ini adalah isi dari line
Surabaya - Kapasan 100000
Ini adalah isi dari line
Jakarta - Harmony 1500000
Ini adalah isi dari line
Jakarta - Monas 950000
Ini adalah isi dari line
Surabaya - Semolowaru 350000
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK a... (yesterday 9:24 P 18:23 LF UTF-8 2 spaces


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help RevenueAggregationJob
RevenueAggregationJob run-job.sh
Add Configuration...
RevenueAggregationMapper.java
RevenueAggregationReducer.java
run-job.sh
Terminal: Local
Version Control Run TODO Problems Terminal Services Profiler Build Dependencies

```
Isi dari values saat ini: 950000
Isi dari values saat ini: 1500000
----------------
Ini dari reducer
Key: Malang
----------------
Isi dari values saat ini: 750000
Isi dari values saat ini: 500000
----------------
Ini dari reducer
Key: Surabaya
----------------
Isi dari values saat ini: 350000
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK a... (yesterday 9:24 P 18:23 LF UTF-8 2 spaces


File Edit View Navigate Code Refactor Build Run Tools VCS Window Help RevenueAggregationJob
RevenueAggregationJob run-job.sh
Add Configuration...
RevenueAggregationMapper.java
RevenueAggregationReducer.java
run-job.sh
Terminal: Local
Version Control Run TODO Problems Terminal Services Profiler Build Dependencies

```
----------------
Ini dari reducer
Key: Jakarta
----------------
Isi dari values saat ini: 1225000
----------------
Ini dari reducer
Key: Malang
----------------
Isi dari values saat ini: 625000
----------------
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built JDK a... (yesterday 9:24 P 18:23 LF UTF-8 2 spaces