TUGAS 1 IF4044
Teknologi Big Data

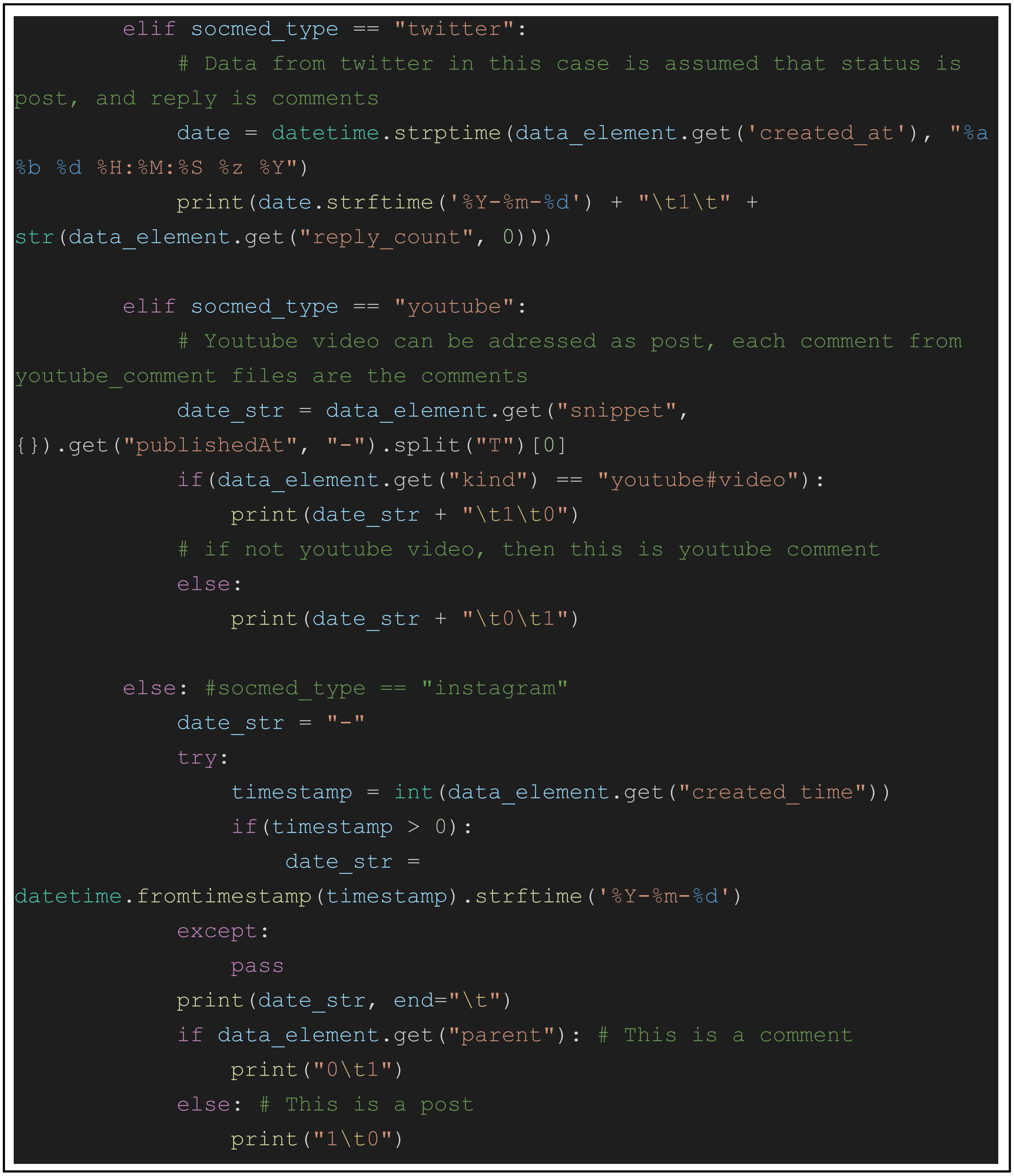
Nama : Kinantan Arya Bagaspati
NIM : 13519044

MapReduce merupakan metode melakukan agregasi data dalam keperluan merangkum atau memvisualisasi data yang lebih memberikan arti/insight terhadap suatu aspek (dalam kasus ini, jumlah post berdasarkan tanggal dan jenis sosmed). MapReduce dilakukan dengan framework Hadoop Mapreduce, pada Virtual Machine bermode Pseudo Distributed yang telah di set up pada milestone lalu. Berikut langkah-langkah pengerjaan/implementasi beserta hasilnya:

1. Download file json dan pelajari dataset. Ternyata nama file json sudah memberikan informasi mengenai tipe sosial media dan tanggalnya, dipastikan tipe sosial media berupa Youtube, Twitter, Instagram, atau Facebook. Akan dipertimbangkan agregat berdasarkan jumlah post atau jumlah komentar.
   Dari tahap ini diperoleh:
    - File berakhiran .json.json tidak digunakan
    - Semua file json merupakan arraylist dari objek dengan format masing-masing, sama setiap jenis sosial media. Tanpa mengetahui nama file, jenis sosial media dapat diketahui dari atribut crawler_target dari data
    - **Facebook** hanya terdiri dari sejenis file, facebook_post_<timestamp>, yang dapat diekstrak informasi dari atribut:
        - Waktu menggunakan created_time
        - List comment menggunakan comments.data
    - **Instagram** terdapat 4 jenis file yakni comment, media, status, dan post. Penulis hanya menganggap comment untuk menambah kalkulasi comment, sedangkan sisanya dihitung sebagai post. Informasi waktu diambil dari atribut created_time
    - **Twitter** hanya terdiri dari sejenis file, twitter_status_<timestamp>, yang dapat diekstraksi:
        - Waktu menggunakan created_time
        - Comments menggunakan reply_count
    - **Youtube** terdiri dari 2 file yakni video dan comments, yang masing-masing berkorespondensi pada kalkulasi post dan comment. Informasi waktu menggunakan atribut createdAt

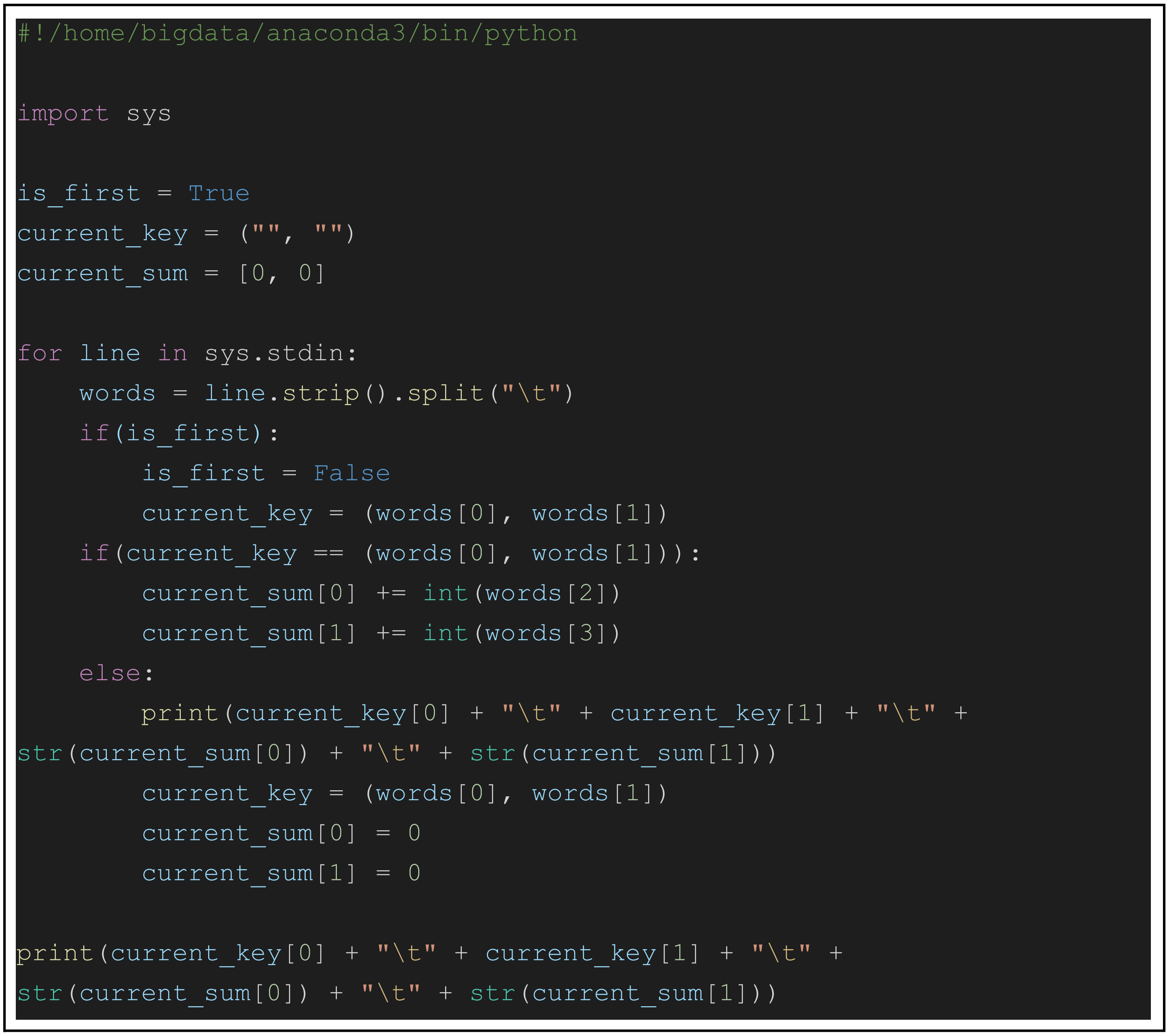
2. Mapper akan disusun sedemikian sehingga menghasilkan tuple berisi informasi tipe sosial media, tanggal, post, dan comment pada setiap entry file

```
#!/home/bigdata/anaconda3/bin/python

import sys
from datetime import datetime
from json import load, loads

#socmed_dict = {}
data_directory = "../raw_json_test"

for line in sys.stdin:
    try:
        data_list = loads(line.strip())
    except:
        continue
    if(type(data_list) is not list):
        continue

    for data_element in data_list:
        # To get the social media type, all json files except from
instagram have this crawler target to differentiate
        socmed_type = "instagram"
        if data_element.get("crawler_target"):
            socmed_type =
data_element["crawler_target"].get("specific_resource_type")
        print(socmed_type, end="\t")

        if socmed_type == "facebook":
            # Data from facebook have its own comments to a post listed
with each own created time
            print(data_element.get("created_time", "-").split("T")[0] +
"\t1\t0")
            for comment in data_element.get("comments", {}).get("data",
[]):
                print(socmed_type + "\t" + comment.get("created_time",
"-").split("T")[0] + "\t0\t1")
```



```
        elif socmed_type == "twitter":
            # Data from twitter in this case is assumed that status is 
post, and reply is comments
            date = datetime.strptime(data_element.get('created_at'), "%a 
%b %d %H:%M:%S %z %Y")
            print(date.strftime('%Y-%m-%d') + "\t1\t" + 
str(data_element.get("reply_count", 0)))

        elif socmed_type == "youtube":
            # Youtube video can be adressed as post, each comment from 
youtube_comment files are the comments
            date_str = data_element.get("snippet", 
{}).get("publishedAt", "-").split("T")[0]
            if(data_element.get("kind") == "youtube#video"):
                print(date_str + "\t1\t0")
            # if not youtube video, then this is youtube comment
            else:
                print(date_str + "\t0\t1")

        else: #socmed_type == "instagram"
            date_str = "-"
            try:
                timestamp = int(data_element.get("created_time"))
                if(timestamp > 0):
                    date_str = 
datetime.fromtimestamp(timestamp).strftime('%Y-%m-%d')
            except:
                pass
            print(date_str, end="\t")
            if data_element.get("parent"): # This is a comment
                print("0\t1")
            else: # This is a post
                print("1\t0")
```

3. Reducer akan disusun sedemikian sehingga menggabungkan hasil mapper berdasarkan jenis sosial media dan tanggal. Penggabungan ini dilakukan pada jumlah post dan jumlah komentar. Hasil ini dapat di print atau sudah langsung dimasukkan pada sebuah file csv.

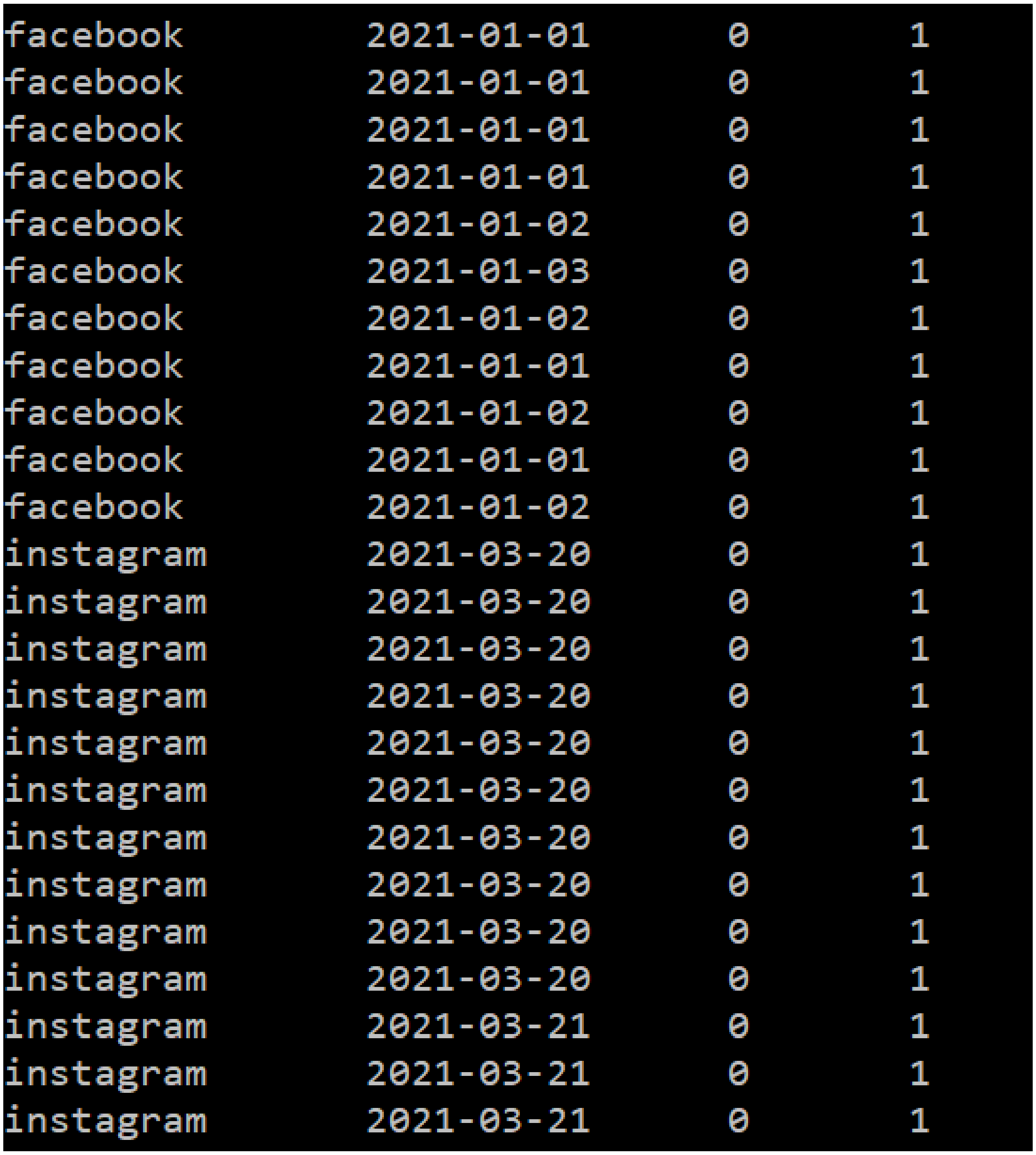


```
#!/home/bigdata/anaconda3/bin/python

import sys

is_first = True
current_key = ("", "")
current_sum = [0, 0]

for line in sys.stdin:
    words = line.strip().split("\t")
    if(is_first):
        is_first = False
        current_key = (words[0], words[1])
    if(current_key == (words[0], words[1])):
        current_sum[0] += int(words[2])
        current_sum[1] += int(words[3])
    else:
        print(current_key[0] + "\t" + current_key[1] + "\t" +
str(current_sum[0]) + "\t" + str(current_sum[1]))
        current_key = (words[0], words[1])
        current_sum[0] = 0
        current_sum[1] = 0

print(current_key[0] + "\t" + current_key[1] + "\t" +
str(current_sum[0]) + "\t" + str(current_sum[1]))
```

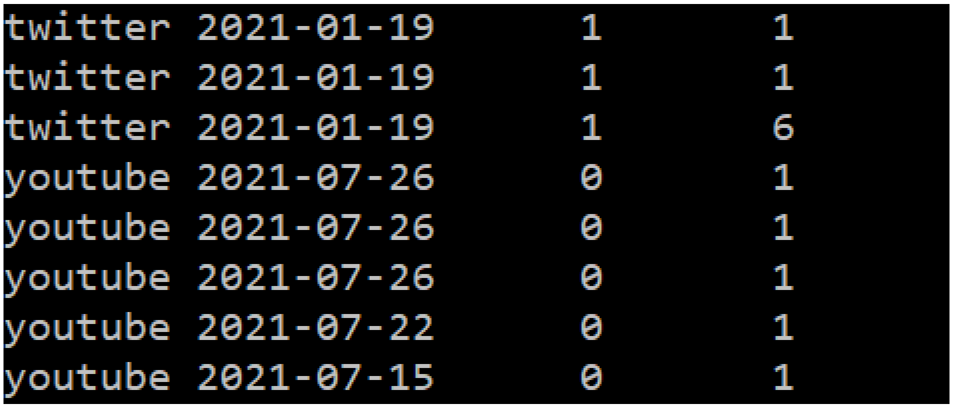
4. Kemudian setelah diujikan dalam lokal (apabila dirasa terlalu memakan waktu, cukup sebagian file json saja yang digunakan), data, mapper, dan reducer akan diuji dalam lingkungan hdfs. Perintah yang digunakan diantaranya:
    - Mengcopy file dari lokal ke hdfs dan sebaliknya menggunakan “hdfs dfs -put <fileloc1> <fileloc2>”
    - Menjalankan job MapReduce dengan perintah:
      hadoop jar /home/bigdata/hadoop-3.2.2/share/hadoop/tools/lib/hadoop-streaming-3.2.2.jar -input <folder data> -output <folder output> -mapper <fileloc mapper> -reducer <fileloc reducer>

Berikut repository untuk versi terupdate dari mapper dan reducer dalam bahasa python: https://github.com/kinantanbagaspati/Tugas1-IF4044

Screenshot hasil perantara mapper sebelum masuk reducer



```
facebook        2021-01-01      0       1
facebook        2021-01-01      0       1
facebook        2021-01-01      0       1
facebook        2021-01-01      0       1
facebook        2021-01-02      0       1
facebook        2021-01-03      0       1
facebook        2021-01-02      0       1
facebook        2021-01-01      0       1
facebook        2021-01-02      0       1
facebook        2021-01-01      0       1
facebook        2021-01-02      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-20      0       1
instagram       2021-03-21      0       1
instagram       2021-03-21      0       1
instagram       2021-03-21      0       1
```



```
twitter 2021-01-19      1       1
twitter 2021-01-19      1       1
twitter 2021-01-19      1       6
youtube 2021-07-26      0       1
youtube 2021-07-26      0       1
youtube 2021-07-26      0       1
youtube 2021-07-22      0       1
youtube 2021-07-15      0       1
```

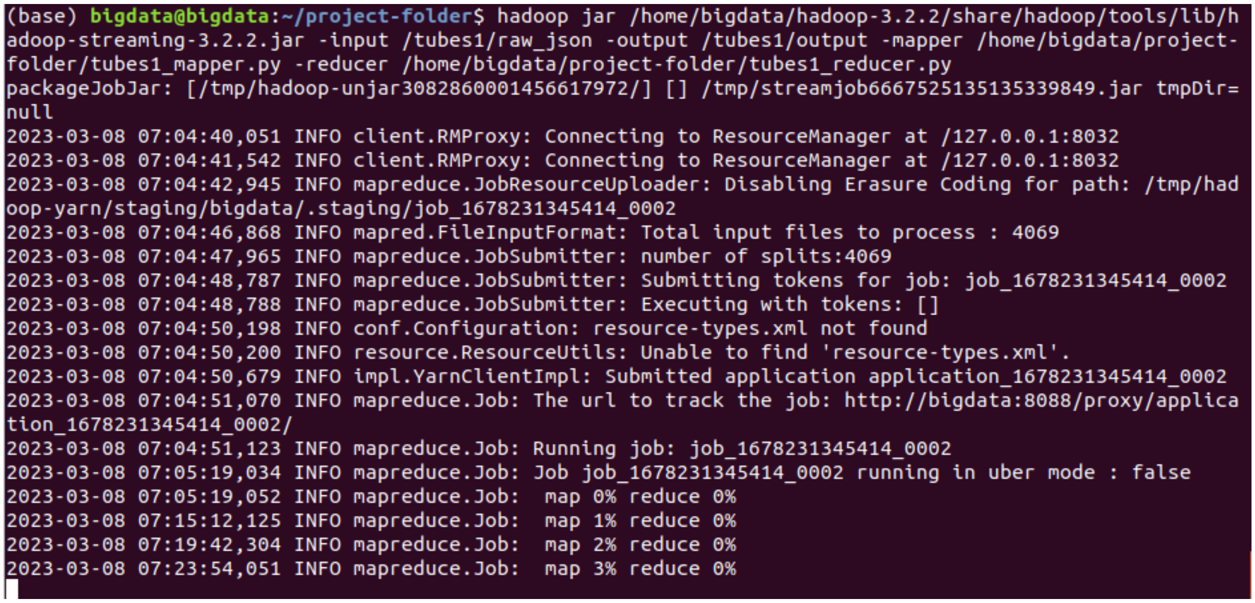
Bukti selesainya hadoop jar command (mapreduce) pada folder test yang berisi tiap jenis file

```
hadoop jar
/home/bigdata/hadoop-3.2.2/share/hadoop/tools/lib/hadoop-streaming-3.2.2.jar -input
/tubes1/raw_json_test -output /tubes1/output_test -mapper
/home/bigdata/project-folder/tubes1_mapper.py -reducer
/home/bigdata/project-folder/tubes1_reducer.py
packageJobJar: [/tmp/hadoop-unjar4158514519586893079/] []
/tmp/streamjob2250570576597211063.jar tmpDir=null
2023-03-08 06:33:30,161 INFO client.RMProxy: Connecting to ResourceManager at
/127.0.0.1:8032
2023-03-08 06:33:32,896 INFO client.RMProxy: Connecting to ResourceManager at
/127.0.0.1:8032
2023-03-08 06:33:34,497 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding
for path: /tmp/hadoop-yarn/staging/bigdata/.staging/job_1678231345414_0001
2023-03-08 06:33:36,361 INFO mapred.FileInputFormat: Total input files to process :
8
2023-03-08 06:33:37,137 INFO mapreduce.JobSubmitter: number of splits:8
2023-03-08 06:33:38,733 INFO mapreduce.JobSubmitter: Submitting tokens for job:
job_1678231345414_0001
2023-03-08 06:33:38,734 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-03-08 06:33:39,999 INFO conf.Configuration: resource-types.xml not found
2023-03-08 06:33:40,000 INFO resource.ResourceUtils: Unable to find
'resource-types.xml'.
2023-03-08 06:33:41,689 INFO impl.YarnClientImpl: Submitted application
application_1678231345414_0001
2023-03-08 06:33:42,174 INFO mapreduce.Job: The url to track the job:
http://bigdata:8088/proxy/application_1678231345414_0001/
2023-03-08 06:33:42,281 INFO mapreduce.Job: Running job: job_1678231345414_0001
2023-03-08 06:35:04,218 INFO mapreduce.Job: Job job_1678231345414_0001 running in
uber mode : false
2023-03-08 06:35:04,261 INFO mapreduce.Job:  map 0% reduce 0%
2023-03-08 06:38:01,558 INFO mapreduce.Job:  map 8% reduce 0%
2023-03-08 06:38:03,042 INFO mapreduce.Job:  map 17% reduce 0%
2023-03-08 06:38:08,265 INFO mapreduce.Job:  map 25% reduce 0%
2023-03-08 06:38:09,270 INFO mapreduce.Job:  map 50% reduce 0%
2023-03-08 06:38:11,353 INFO mapreduce.Job:  map 75% reduce 0%
```

```
2023-03-08 06:41:26,761 INFO mapreduce.Job:  map 88% reduce 0%
2023-03-08 06:41:27,765 INFO mapreduce.Job:  map 100% reduce 0%
2023-03-08 06:41:31,786 INFO mapreduce.Job:  map 100% reduce 100%
2023-03-08 06:41:36,822 INFO mapreduce.Job: Job job_1678231345414_0001 completed
successfully
2023-03-08 06:41:41,494 INFO mapreduce.Job: Counters: 55
        File System Counters
                FILE: Number of bytes read=59877
                FILE: Number of bytes written=2249780
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=2782441
                HDFS: Number of bytes written=10385
                HDFS: Number of read operations=29
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Killed map tasks=2
                Launched map tasks=10
                Launched reduce tasks=1
                Data-local map tasks=10
                Total time spent by all maps in occupied slots (ms)=1793248
                Total time spent by all reduces in occupied slots (ms)=189881
                Total time spent by all map tasks (ms)=1793248
                Total time spent by all reduce tasks (ms)=189881
                Total vcore-milliseconds taken by all map tasks=1793248
                Total vcore-milliseconds taken by all reduce tasks=189881
                Total megabyte-milliseconds taken by all map tasks=1836285952
                Total megabyte-milliseconds taken by all reduce tasks=194438144
        Map-Reduce Framework
                Map input records=8
                Map output records=2268
                Map output bytes=55335
                Map output materialized bytes=59919
                Input split bytes=1095
                Combine input records=0
                Combine output records=0
                Reduce input groups=4
                Reduce shuffle bytes=59919
                Reduce input records=2268
                Reduce output records=428
                Spilled Records=4536
                Shuffled Maps =8
                Failed Shuffles=0
                Merged Map outputs=8
                GC time elapsed (ms)=10361
                CPU time spent (ms)=22460
                Physical memory (bytes) snapshot=1800249344
                Virtual memory (bytes) snapshot=22811975680
                Total committed heap usage (bytes)=1405714432
                Peak Map Physical memory (bytes)=228376576
```

```
              Peak Map Virtual memory (bytes)=2569568256
              Peak Reduce Physical memory (bytes)=123834368
              Peak Reduce Virtual memory (bytes)=2541072384
       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=2781346
       File Output Format Counters
              Bytes Written=10385
2023-03-08 06:41:41,509 INFO streaming.StreamJob: Output directory:
/tubes1/output_test
```

Bukti keberjalanan MapReduce pada keseluruhan folder (1GB)



```
(base) bigdata@bigdata:~/project-folder$ hadoop jar /home/bigdata/hadoop-3.2.2/share/hadoop/tools/lib/h
adoop-streaming-3.2.2.jar -input /tubes1/raw_json -output /tubes1/output -mapper /home/bigdata/project-
folder/tubes1_mapper.py -reducer /home/bigdata/project-folder/tubes1_reducer.py
packageJobJar: [/tmp/hadoop-unjar3082860001456617972/] [] /tmp/streamjob6667525135135339849.jar tmpDir=
null
2023-03-08 07:04:40,051 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2023-03-08 07:04:41,542 INFO client.RMProxy: Connecting to ResourceManager at /127.0.0.1:8032
2023-03-08 07:04:42,945 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/had
oop-yarn/staging/bigdata/.staging/job_1678231345414_0002
2023-03-08 07:04:46,868 INFO mapred.FileInputFormat: Total input files to process : 4069
2023-03-08 07:04:47,965 INFO mapreduce.JobSubmitter: number of splits:4069
2023-03-08 07:04:48,787 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1678231345414_0002
2023-03-08 07:04:48,788 INFO mapreduce.JobSubmitter: Executing with tokens: []
2023-03-08 07:04:50,198 INFO conf.Configuration: resource-types.xml not found
2023-03-08 07:04:50,200 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2023-03-08 07:04:50,679 INFO impl.YarnClientImpl: Submitted application application_1678231345414_0002
2023-03-08 07:04:51,070 INFO mapreduce.Job: The url to track the job: http://bigdata:8088/proxy/applica
tion_1678231345414_0002/
2023-03-08 07:04:51,123 INFO mapreduce.Job: Running job: job_1678231345414_0002
2023-03-08 07:05:19,034 INFO mapreduce.Job: Job job_1678231345414_0002 running in uber mode : false
2023-03-08 07:05:19,052 INFO mapreduce.Job:  map 0% reduce 0%
2023-03-08 07:15:12,125 INFO mapreduce.Job:  map 1% reduce 0%
2023-03-08 07:19:42,304 INFO mapreduce.Job:  map 2% reduce 0%
2023-03-08 07:23:54,051 INFO mapreduce.Job:  map 3% reduce 0%
```

SQL Equivalent

```
CREATE TABLE Posts (
    SocMedType VARCHAR(16),
    PostID VARCHAR(255),
    CreatedAt DATE,
    ...
    PRIMARY KEY (PostID)
);
CREATE TABLE Comments (
    PostID VARCHAR(255),
    CommentID VARCHAR(255),
    CreatedAt DATE,
    ...
    PRIMARY KEY (CommentID),
    FOREIGN KEY PostID References Posts(PostID)
);
CREATE TABLE Replies (
    CommentID VARCHAR(255),
    ReplyID VARCHAR(255),
    CreatedAt DATE,
    ...
    PRIMARY KEY (ReplyID),
    FOREIGN KEY PostID References Comment(CommentID)
);

WITH CommentAgged AS (
    SELECT PostID, COUNT(ReplyID)+1 as ReplyCount
    FROM Comments JOIN Replies
    ON Comments.CommentID = Replies.CommentID
    GROUP BY Comments.CommentID
),
SELECT SocMedType, Posts.CreatedAt, Count(PostID) as PostCount,
Count(ReplyCount) as CommentCount
FROM Posts JOIN CommentAgged
ON Posts.PostID = CommentAgged.PostID
GROUP BY SocMedType, Posts.CreatedAt
```

Kesimpulan dan Saran

- Hardware pada pengerjaan tugas ini sangat berpengaruh. Keterbatasan VM menyebabkan script python biasa untuk melakukan MapReduce masih jauh lebih cepat ketimbang Hadoop Jar (Hitungan detik vs hitungan hari)
- Penyusunan script untuk MapReduce sangat menguntungkan kasus yang butuh pendekatan prosedural
- Banyak visualisasi lain yang dapat dicapai, misalkan hubungan waktu post/komen dengan scraping (dari nama file), keterikatan tiap komen dengan tiap post untuk melihat traksi, hubungan pengguna antar platform social media, dan lain sebagainya