

**LAPORAN**  
**RENCANA TUGAS MANDIRI (RTM) Ke-3**  
**MATA KULIAH BIG DATA**  
**“IMPLEMENTASI PROGRAM MAPREDUCE**  
**WORDCOUNT DI HDFS DAN PYTHON”**



**DISUSUN OLEH:**

Citra Amelia Intan Permadani (21083010004)

**DOSEN PENGAMPU:**

Tresna Maulana Fahrudin S.ST., M.T. (NIP. 199305012022031007)

**PROGRAM STUDI SAINS DATA**  
**FAKULTAS ILMU KOMPUTER**  
**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR**  
**2023**

# IMPLEMENTASI

## A. HDFS

1. Buat direktori di home oracle dengan fungsi *mkdir* dan nama folder *tugas\_wordcount*.

```
oracle@bigdatalite:~  
File Edit View Search Terminal Help  
[oracle@bigdatalite ~]$ mkdir tugas_wordcount
```

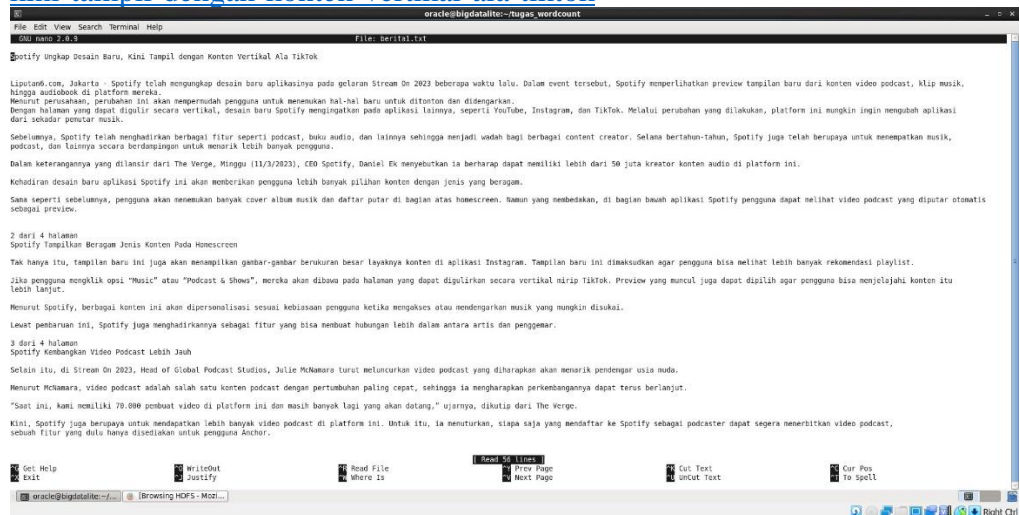
2. Masuk ke folder *tugas\_wordcount* dengan fungsi *cd*. Maka folder akan terbuka dengan bukti di bagian kiri terdapat nama folder *tugas\_wordcount*.

```
[oracle@bigdatalite ~]$ cd tugas_wordcount  
[oracle@bigdatalite tugas_wordcount]$
```

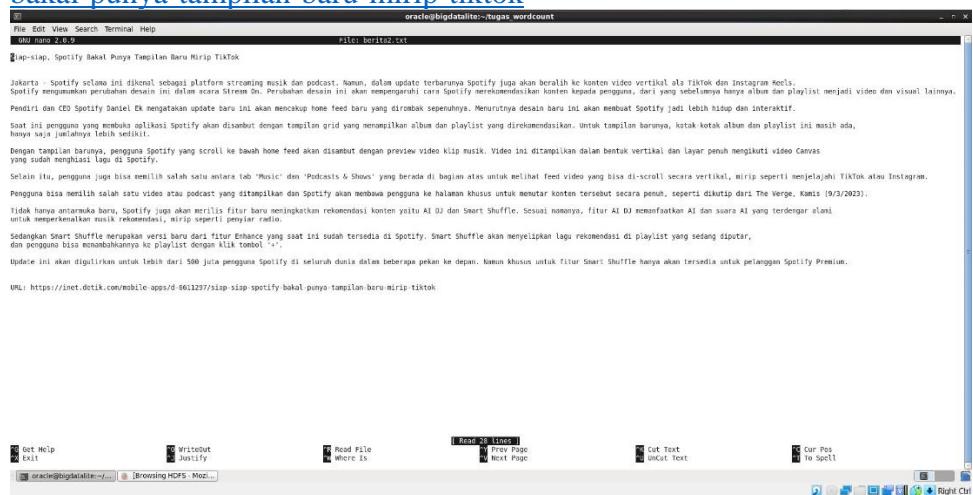
3. Buat file txt dengan fungsi *nano* yang berisi 2 berita dengan topik yang sama tetapi dengan sumber yang berbeda. Disini saya memakai topik

- Spotify Ungkap Desain Baru, Kini Tampil dengan Konten Vertikal Ala TikTok bersumber dari liputan6

<https://www.liputan6.com/tekno/read/5229410/spotify-ungkap-desain-baru-kini-tampil-dengan-konten-vertikal-ala-tiktok>



- Siap-siap, Spotify Bakal Punya Tampilan Baru Mirip TikTok bersumber dari detik.com <https://inet.detik.com/mobile-apps/d-6611297/siap-siap-spotify-bakal-punya-tampilan-baru-mirip-tiktok>



Dari 2 sumber berita di atas, *berita1.txt* menyimpan isi berita sumber liputan6 dan *berita2.txt* menyimpan isi berita sumber detik.com

```
[oracle@bigdatalite tugas_wordcount]$ nano berita1.txt
[oracle@bigdatalite tugas_wordcount]$ nano berita2.txt
```

4. Kita copy file WordCount.java ke folder tugas\_wordcount dengan fungsi *cp*, pertama-tama kita keluar dulu dari direktori tugas\_wordcount karena file WordCount.java berada di home.

```
[oracle@bigdatalite tugas_wordcount]$ cd
[oracle@bigdatalite ~]$ cp WordCount.java tugas_wordcount
```

5. Masuk lagi ke direktori tugas\_wordcount lalu kita export file tools jar ke classpath yang digunakan oleh Hadoop.

Classpath adalah daftar lokasi file JAR dan direktori yang berisi file kelas Java yang digunakan oleh sebuah program Java. Dalam konteks Hadoop, classpath digunakan untuk memuat library dan file konfigurasi yang diperlukan oleh Hadoop dan aplikasi yang dijalankan di atasnya.

```
[oracle@bigdatalite ~]$ cd tugas_wordcount
[oracle@bigdatalite tugas_wordcount]$ export HADOOP_CLASSPATH=/usr/java/jdk1.8.0_151/lib/tools.jar
```

Tuliskan script '\$ hadoop com.sun.tools.javac.Main WordCount.java' ini yang digunakan untuk mengkompilasi file 'WordCount.java' menjadi file 'WordCount.class'. Dalam hal ini, perintah hadoop digunakan untuk menjalankan kompiler 'javac' yang ada di dalam direktori 'tools.jar' yang telah ditambahkan ke classpath sebelumnya.

```
[oracle@bigdatalite tugas_wordcount]$ hadoop com.sun.tools.javac.Main WordCount.java
```

Setelah berhasil maka tuliskan script '\$ jar cf wc.jar WordCount\*.class' yang digunakan untuk mengemas file 'WordCount.class' menjadi sebuah file JAR dengan nama 'wc.jar'. Perintah ini menggunakan tool jar yang tersedia di dalam JDK untuk membuat file JAR.

```
[oracle@bigdatalite tugas_wordcount]$ jar cf wc.jar WordCount*.class
```

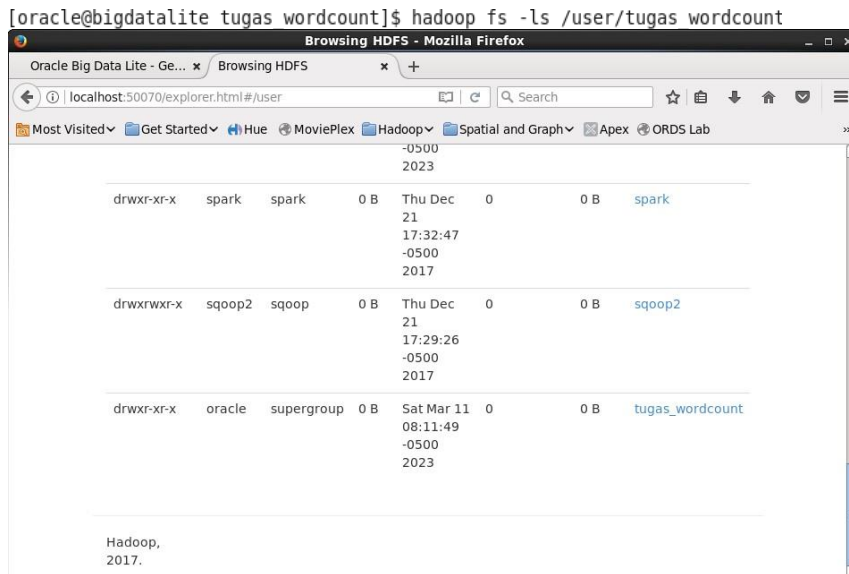
6. Kemudian kita cek apakah file txt dan beberapa file tools jar telah masuk ke dalam direktori tugas\_wordcount dengan fungsi *ls-l*.

```
[oracle@bigdatalite tugas_wordcount]$ ls -l
total 28
-rwxrwxrwx. 1 oracle oinstall 4033 Mar 11 07:58 berita1.txt
-rwxrwxrwx. 1 oracle oinstall 2486 Mar 11 08:02 berita2.txt
-rw-r--r--. 1 oracle oinstall 3075 Mar 11 08:10 wc.jar
-rw-r--r--. 1 oracle oinstall 1491 Mar 11 08:10 WordCount.class
-rw-r--r--. 1 oracle oinstall 1739 Mar 11 08:10 WordCount$IntSumReducer.class
-rw-r--r--. 1 oracle oinstall 2089 Mar 11 08:09 WordCount.java
-rw-r--r--. 1 oracle oinstall 1736 Mar 11 08:10 WordCount$TokenizerMapper.class
[oracle@bigdatalite tugas_wordcount]$
```

7. Selanjutnya kita telah tersambung dengan direktori di Hadoop sehingga kita dapat membuat direktori di hadoop dengan *hadoop fs -mkdir*

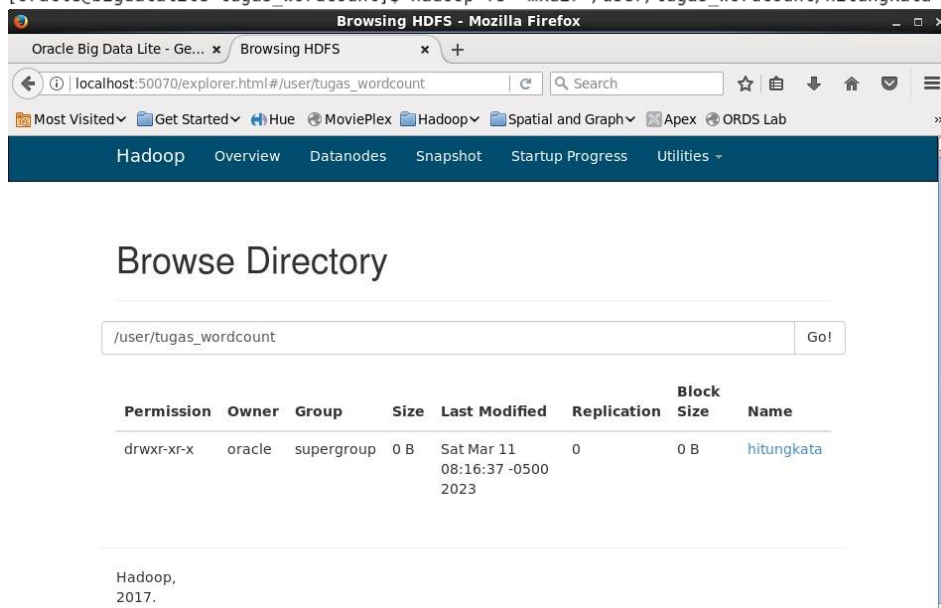
```
[oracle@bigdatalite tugas_wordcount]$ hadoop fs -mkdir hdfs:///user/tugas_wordcount
```

Kita cek direktori yang telah dibuat apakah telah terbentuk dengan *hadoop fs -ls*. Sebelumnya kita buka terlebih dahulu *localhost:50070* untuk membuka direktori hadoop di browser. Dan pastikan bahwa direktori tugas\_wordcount telah ada di daftar user di browser hadoop seperti di bawah ini.



Di dalam direktori tugas\_wordcount dalam hadoop kita buat direktori baru dengan nama hitungkata

```
[oracle@bigdatalite tugas_wordcount]$ hadoop fs -mkdir /user/tugas_wordcount/hitungkata
```

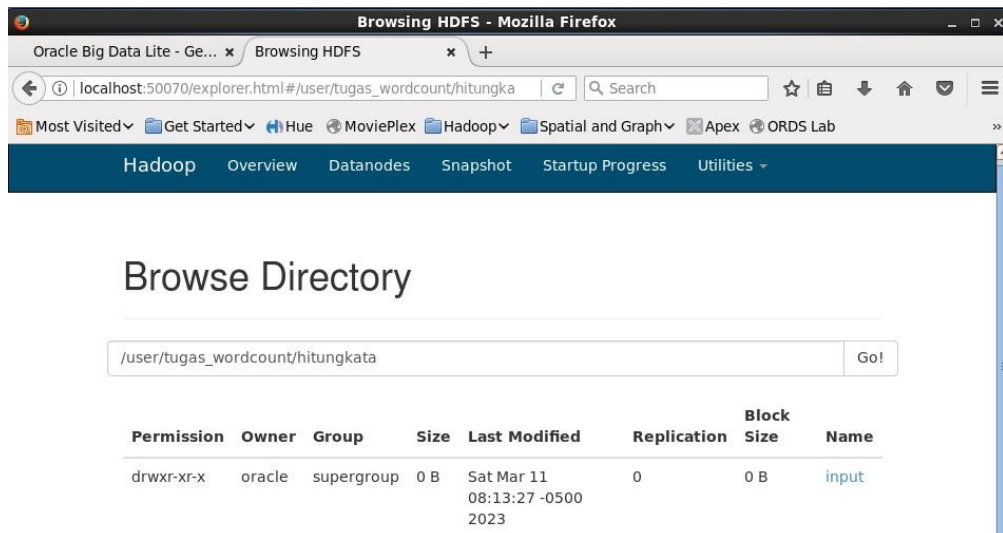


lalu buat lagi direktori di dalam direktori hitung kata dengan nama input

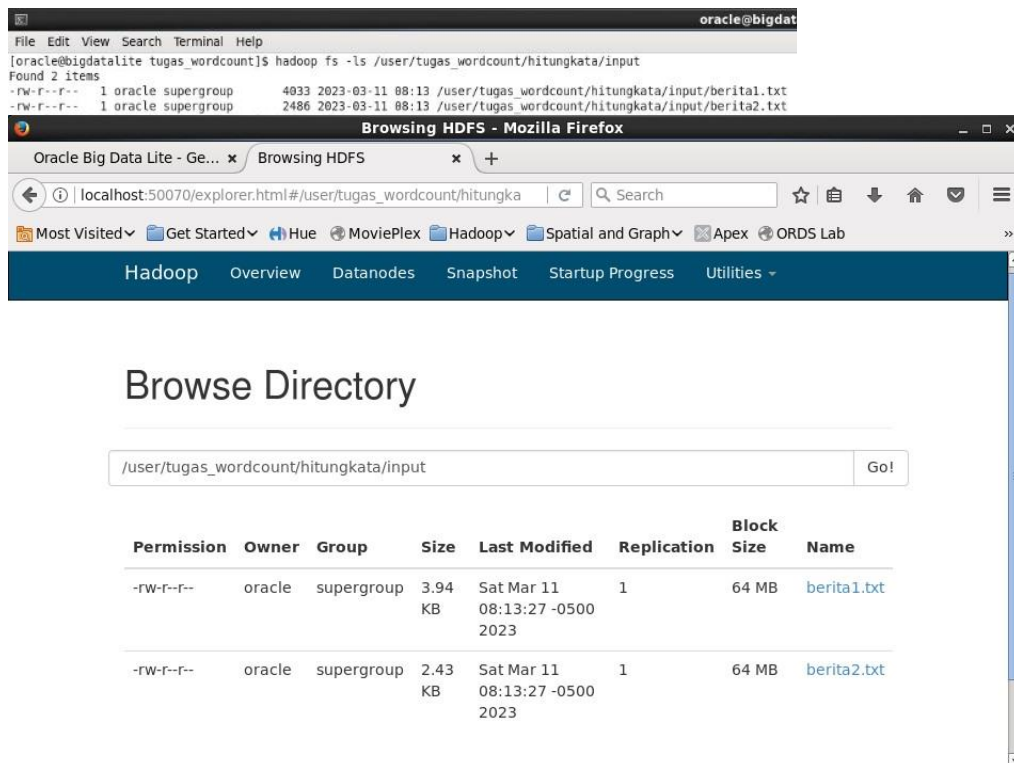
```
[oracle@bigdatalite tugas_wordcount]$ hadoop fs -mkdir /user/tugas_wordcount/hitungkata/input
```

yang isinya nanti 2 file txt berita yang telah dibuat di oracle caranya dengan meng-copy from local berita\* (\* artinya terdapat lebih dari 1 file dengan nama berita).

```
[oracle@bigdatalite tugas_wordcount]$ hadoop fs -copyFromLocal berita* /user/tugas_wordcount/hitungkata/input
```



8. Kita juga dapat melihat apakah 2 file berita txt sudah masuk ke direktori input dalam Hadoop dengan '\$ *hadoop fs -ls* ....' maka akan ditemukan 2 items berita 2 file txt berita.



Kita juga bisa melihat isi teks berita.txt dengan '\$ *hadoop fs -cat* /user/tugas\_wordcount/hitungkata/input/berita2.txt' Seperti di bawah ini

```
[oracle@bigdata1ite tugas_wordcount]$ hadoop fs -cat /user/tugas_wordcount/hitungkata/input/berita2.txt
Siap-siap, Spotify Bakal Punya Tampilan Baru Mirip TikTok
```

Jakarta - Spotify selama ini dikenal sebagai platform streaming musik dan podcast. Namun, dalam update terbarunya Spotify juga akan beralih ke konten video vertikal ala TikTok dan Instagram Reels. Spotify mengumumkan perubahan desain ini dalam acara Stream On. Perubahan desain ini akan mempengaruhi cara Spotify merekomendasikan konten kepada pengguna, dari yang sebelumnya hanya album dan playlist menjadi video dan visual lainnya.

Pendiri dan CEO Spotify Daniel Ek mengatakan update baru ini akan mencakup home feed baru yang dirombak sepenuhnya. Menurutnya desain baru ini akan membuat Spotify jadi lebih hidup dan interaktif.

Saat ini pengguna yang membuka aplikasi Spotify akan disambut dengan tampilan grid yang menampilkan album dan playlist yang direkomendasikan. Untuk tampilan barunya, kotak-kotak album dan playlist ini masih ada, hanya saja jumlahnya lebih sedikit.

Dengan tampilan barunya, pengguna Spotify yang scroll ke bawah home feed akan disambut dengan preview video klip musik. Video ini ditampilkan dalam bentuk vertikal dan layar penuh mengikuti video Canvas yang sudah menghiasi lagu di Spotify.

Selain itu, pengguna juga bisa memilih salah satu antara tab 'Music' dan 'Podcasts & Shows' yang berada di bagian atas untuk melihat feed video yang bisa di-scroll secara vertikal, mirip seperti menjelajahi TikTok atau Instagram.

Pengguna bisa memilih salah satu video atau podcast yang ditampilkan dan Spotify akan membawa pengguna ke halaman khusus untuk memutar konten tersebut secara penuh, seperti dikutip dari The Verge, Kamis (9/3/2023).

Tidak hanya antarmuka baru, Spotify juga akan merilis fitur baru meningkatkan rekomendasi konten yaitu AI DJ dan Smart Shuffle. Sesuai namanya, fitur AI DJ memanfaatkan AI dan suara AI yang terdengar alami untuk memperkenalkan musik rekomendasi, mirip seperti penyiar radio.

Sedangkan Smart Shuffle merupakan versi baru dari fitur Enhance yang saat ini sudah tersedia di Spotify. Smart Shuffle akan menyelipkan lagu rekomendasi di playlist yang sedang diputar, dan pengguna bisa menabuhkannya ke playlist dengan klik tombol "+".

Update ini akan digulirkan untuk lebih dari 500 juta pengguna Spotify di seluruh dunia dalam beberapa pekan ke depan. Namun khusus untuk fitur Smart Shuffle hanya akan tersedia untuk pelanggan Spotify Premium.

URL: <https://inet.detik.com/mobile-apps/d-6611297/siap-siap-spotify-bakal-punya-tampilan-baru-mirip-tiktok>



9. Lalu kita jalankan JAR dengan '\$ *hadoop jar wc.jar WordCount /user/hadoop1/hitungkata/input /user/hadoop1/hitungkata/output* ' untuk menjalankan sebuah job MapReduce dengan menggunakan file JAR wc.jar yang telah dibuat sebelumnya dan class WordCount yang ada di dalamnya. Dimana file berita dari direktori input akan dijalankan dan hasilnya akan disimpan pada direktori output.

Program akan berjalan hingga map dan reduce menghasilkan 100% dan muncul **completed successfully**

```
oracle@bigdatalite: tugas_wordcount$ hadoop jar wc.jar WordCount /user/tugas_wordcount/hitungkata/input /user/tugas_wordcount/hitungkata/output
23/03/11 08:16:29 INFO client.NMProxy: Connecting to ResourceManager at /0.0.0.0:8032
23/03/11 08:16:30 WARN mapreduce.JobSourceImpl: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
23/03/11 08:16:30 INFO input.FileInputFormat: Total input paths to process : 2
23/03/11 08:16:30 INFO mapreduce.JobSubmitter: number of splits:2
23/03/11 08:16:30 INFO mapreduce.JobSubmitter: Submitting tokens for job: Job_1678533389132_0001
23/03/11 08:16:31 INFO impl.YarnClientImpl: Submitted application application_1678533389132_0001
23/03/11 08:16:31 INFO mapreduce.Job: The url to track the job: http://bigdatalite.localdomain:8088/proxy/application_1678533389132_0001/
23/03/11 08:16:31 INFO mapreduce.Job: Running job: Job_1678533389132_0001
23/03/11 08:16:38 INFO mapreduce.Job: Job job_1678533389132_0001 running in uber mode : false
23/03/11 08:16:44 INFO mapreduce.Job: map 0% reduce 0%
23/03/11 08:16:45 INFO mapreduce.Job: map 100% reduce 0%
23/03/11 08:16:50 INFO mapreduce.Job: map 100% reduce 100%
23/03/11 08:16:51 INFO mapreduce.Job: Job job_1678533389132_0001 completed successfully
```

The screenshot displays the Hadoop job completion logs and the HDFS browser interface. The logs show the job 'Job\_1678533389132\_0001' completed successfully. The HDFS browser shows the directory structure and file details.

**File System Counters**

Counter	Value
FILE: Number of bytes read	6908
FILE: Number of bytes written	449685
FILE: Number of read operations	0
FILE: Number of large read operations	0
FILE: Number of write operations	0
HDFS: Number of bytes read	4619
HDFS: Number of bytes written	4247
HDFS: Number of read operations	9
HDFS: Number of large read operations	0
HDFS: Number of write operations	2

**Job Counters**

Counter	Value
Launched map tasks	2
Launched reduce tasks	2
Data-local map tasks	2
Total time spent by all maps in occupied slots (ms)	8582
Total time spent by all reduces in occupied slots (ms)	2506
Total time spent by all map tasks (ms)	8582
Total time spent by all reduce tasks (ms)	2506
Total vcore-milliseconds taken by all map tasks	8582
Total vcore-milliseconds taken by all reduce tasks	2506
Total megabyte-milliseconds taken by all map tasks	8787968
Total megabyte-milliseconds taken by all reduce tasks	2566144

**Map-Reduce Framework**

Counter	Value
Map input records	84
Map output records	888
Map output bytes	9981
Map output materialized bytes	6914
Input split bytes	300
Combine input records	888
Combine output records	486
Reduce input groups	484
Reduce shuffle bytes	6914
Reduce input records	486
Reduce output records	484
Spilled Records	972
Shuffled Maps	2
Failed Shuffles	0
Merged Map outputs	2
GC time elapsed (ms)	200
CPU time spent (ms)	2888
Physical memory (bytes) snapshot	738648064
Virtual memory (bytes) snapshot	6288314368
Total committed heap usage (bytes)	493879296

**Shuffle Errors**

Error	Count
BAD ID=0	0
CONNECTION=0	0
IO_ERROR=0	0
WRONG_LENGTH=0	0
WRONG_MAP=0	0
WRONG_REDUCE=0	0

**File Input Format Counters**

Counter	Value
Bytes Read	6519
File Output Format Counters	0
Bytes Written	4247

**Browsing HDFS - Mozilla Firefox**

Oracle Big Data Lite - Ge... x Browsing HDFS x +

localhost:50070/explorer.html#/user/tugas\_wordcount/hitungkata

Most Visited Get Started Hue MoviePlex Hadoop Spatial and Graph Apex ORDS Lab

Hadoop Overview Datanodes Snapshot Startup Progress Utilities

### Browse Directory

/user/tugas\_wordcount/hitungkata Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	oracle	supergroup	0 B	Sat Mar 11 08:13:27 -0500 2023	0	0 B	input
drwxr-xr-x	oracle	supergroup	0 B	Sat Mar 11 08:16:48 -0500 2023	0	0 B	output

10. Dan yang terakhir kita dapat memunculkan hasil jumlah kata setiap kata dalam setiap file berita dengan

'\$hadoop fs -cat /pengguna/hadoop1/hitungkata/output/part\*'

```
[oracle@bigdatalite tugas_wordcount]$ hadoop fs -cat /user/tugas_wordcount/hitungkata/output/part*
"Music" 1
6 2
+,. 1
"Music" 1
"Podcasts" 1
(11/3/2023), 1
(9/3/2023), 1
2 2
2023 1
2023, 1
3 1
4 4
56 1
569 1
70.960 1
AI 4
Ala 1
Anchor 1
Bakal 1
Baru 2
Baru, 1
Beragam 1
CEO 2
Canvas 1
DJ 2
Dolan 2
Daniel 2
Dengan 2
Desain 2
Di 1
EK 2
Enhance 1
Global 1
Head 1
Homescreen 1
Instagram 1
Instagram, 1
Instagram, 2
Jakarta 2
Jauh 1
Jenis 1
Jika 1
Julie 1
Kamis 1
Kehadiran 1
Kembangkan 1
```

oracle@bigdatalite:~/... [Browsing HDFS - Mozi...

Dan hasil yang di dapatkan adalah

File Edit View Search Terminal Help						orach						File Edit View Search Terminal Help						oracle@bigdatalite:/tugas					
File Output Format Counters						File Edit View Search Terminal Help						File Edit View Search Terminal Help						File Edit View Search Terminal Help					
[oracle@bigdatalite tugas_wordcount]\$ hadoop fs -cat /user/tugas_wordcount/hitungkata/output/part*						Kamis 1						Tak 1						Tampil 1					
"Music" 1						Kehadiran 1						Tampilan 3						Tampilkan 1					
6 2						Kembangkan 1						The 3						Tidak 1					
+,. 1						Kini, 1						TikTok 4						TikTok, 2					
"Podcasts" 1						Konten 2						URL: 2						Ungkap 1					
(11/3/2023), 1						Kontradiksi 1						Untuk 2						Update 1					
(9/3/2023), 1						Lebih 1						Verge, 2						Vertikal 1					
2 2						Lewat 1						Video 2						YouTube, 1					
2023 1						Liputan6.com, 1						acara 1						ada, 1					
2023, 1						McNamara 1						adalah 1						agar 2					
3 1						McNamara, 1						akan 21						akses, 1					
4 4						Melalui 1						ala 1						alami 1					
56 1						Menurut 3						album 4						antara 2					
569 1						Menurutnya 1						antarmuka 1						aplikasi 7					
70.960 1						Meskipun 1						aplikasinya 1						artikl 2					
AI 4						Minggu 1						atas 2						atau 4					
Ala 1						Mirip 1						audio 1						audio, 1					
Anchor 1						Namun 3						audiobook 1						bagi 2					
Bakal 1						Namun, 1						bagian 3						banyak 7					
Baru 2						On 2						baru 14						baru, 1					
Baru, 1						On, 1						baranya, 2						bawah 2					
Beragam 1						Pada 1						beberapa 2						belum 1					
CEO 2						Pendiri 1						bentuk 2						berada 1					
Canvas 1						Pengguna 1						berada, 1						berapas, 1					
DJ 2						Perubahan 1						beralih 1						beralih 5					
Dolan 2						Podcast 2																	
Daniel 2						Premium, 1																	
Dengan 2						Previous 1																	
Desain 2						Punya 1																	
Di 1						Revisi 1																	
EK 2						Saat 1																	
Enhance 1						Sama 1																	
Global 1						Sebelumnya, 1																	
Head 1						Sedangkan 1																	
Homescreen 1						Selama 1																	
Instagram 1						Sesuai 1																	
Instagram, 1						Shows 1																	
Instagram, 2						Shows*, 1																	
Jakarta 2						Shuffle 3																	
Jauh 1						Shuffle, 1																	
Jenis 1						Siap-siap, 1																	
Jika 1						Smart 1																	
Julie 1						Spotify 29																	
Kamis 1						Spotify, 2																	
Kehadiran 1						Spotify, 3																	
Kembangkan 1						Stream 3																	
						Studios, 1																	
						Tak 1																	

File	Edit	View	Search	Terminal	Help	File	Edit	View	Search	Terminal	Help	File	Edit	View	Search	Terminal	Help	File	Edit	View	Search	Terminal	Help
lagu 2						menungkap 1						podcast 6						sedang 1					
lain 1						meningkatkan 1						podcast, 4						sedikit 1					
lainnya 2						menjadi 2						podcast, 1						segera 1					
lainnya, 1						menjelajahi 2						podcaster 1						sehingga 2					
lahu 1						menuturkan, 1						preview 2						sebagai 1					
lanjut, 1						menyebutkan 1						putar 1						selama 1					
layaknya 1						menyimpulkan 1						radio, 1						seluruh 1					
layar 1						nereka 4						rekendasi 3						sepemunya, 1					
lebih 11						nereka, 1						rekendasi, 1						seperti 6					
masih 2						nerekomendasikan 1						saat 1						sering 1					
melebakkan 1						nerilis 1						saja 2						sebagai 1					
melihat 3						nerupakan 2						salah 3						sebagai 1					
meluncurkan 1						nirip 3						satu 3						streaming 2					
memanfaatkan 1						muds, 1						scroll 1						suara 1					
membawa 1						muncul 1						sebagai 4						sudah 2					
membedakan, 1						mungkin 4						sebagai 2						sukai 1					
memberikan 2						musik 4						sebelumnya 1						tab 1					
membut 2						musik, 3						sebelumnya, 1						tampilan 6					
menbuka 1						namanya, 1						sebuah 1						telah 4					
menilih 2						of 1						secara 2						tentu 1					
memiliki 2						opsi 1						sedang 1						terbaruanya 1					
menpengaruhi 1						otomatis 1						sedikit, 1						terbesar 1					
memperkenalkan 1						pada 3						segera 1						terbiasa 1					
menperlihatkan 1						paling 1						sehingga 2						terdengar 1					
menpermudah 1						pekan 1						sebagai 1						terlihat 1					
menutar 1						pelanggan 1						selama 1						tersebut 1					
menandukannya 1						peabaruhan 2						seperunya, 1						tersebut, 1					
menampilan 2						pebuat 1						seperunya, 1						tersedia 2					
menarik 2						pebutar 1						sering 1						terus 2					
menacup 1						pendengar 1						sebagai 1						tetapi 1					
mendafar 1						penggemar, 1						sebagai 1						tomboh 1					
mendapatkan 1						pengguna 18						sebagai 1						turut 1					
mendengarkan 1						pengguna, 2						sisi 1						ujarnya, 1					
menelusuri 1						pengguna, 1						streaming 2						untuk 13					
menenapkan 1						pengguna, 1						suara 1						update 2					
menemukan 3						penub 1						sudah 2						usia 1					
menertikan 1						penub, 1						sukai 1						utama 1					
menekses 1						penyar 1						tab 1						versi 1					
menatakan 1						perkembangannya 1						tampilan 6						vertikal 3					
menhadirkan 1						pertumbuhan 1						telah 4						vertikal, 2					
menghadirkannya 1						perubahan 4						tentu 1						video 13					
mengharapkan 1						perusahaan, 1						terbaruanya 1						visual 1					
menghass 1						pilihan 2						terbesar 1						wadah 1					
mengikuti 1						platform 7						terbiasa 1						waktu 1					
mengingatkan 1						playlist 5						terdengar 1						yaitu 1					
menklik 1						playlist, 1						terlihat 1						yang 31					
mengubah 1						podcast 6						tersebut 1						*Podcast 1					
menggunakan 1						podcast, 4						tersebut, 1						[oracle@bigdatalite tugas_wordcount]\$					
menungkap 1						podcaster 1						tersedia 2											

Dan dalam Hadoop Browser terlihat seperti di bawah ini yang menyatakan bahwa program MapReduce WordCount telah Success

Browsing HDFS - Mozilla Firefox

Oracle Big Data Lite - Ge...

Browsing HDFS

localhost:50070/explorer.html#/user/tugas\_wordcount/hitungkata

Search

Most Visited

Get Started

Hue

MoviePlex

Hadoop

Spatial and Graph

Apex

ORDS Lab

Hadoop

Overview

Datanodes

Snapshot

Startup Progress

Utilities

Browse Directory

/user/tugas\_wordcount/hitungkata/output

Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	oracle	supergroup	0 B	Sat Mar 11 08:16:48 -0500 2023	1	64 MB	<a href="#">_SUCCESS</a>
-rw-r--r--	oracle	supergroup	4.15 KB	Sat Mar 11 08:16:48 -0500 2023	1	64 MB	<a href="#">part-r-00000</a>

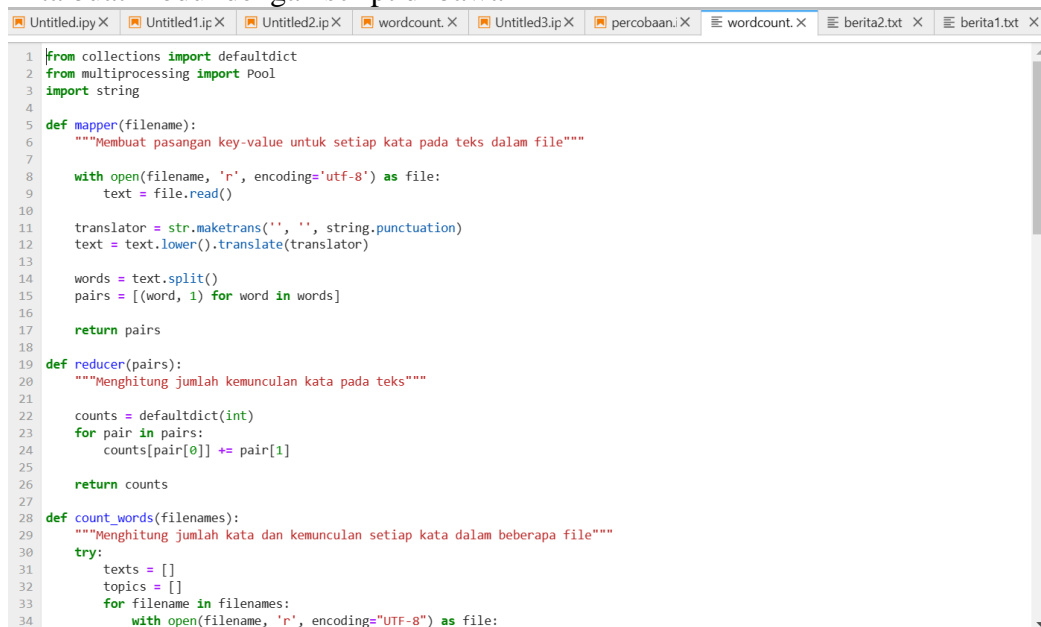


## B. PYTHON

1. Buat file txt dengan fungsi *nano* yang berisi 2 berita dengan topik yang sama tetapi dengan sumber yang berbeda. Disini saya memakai topik
  - Spotify Ungkap Desain Baru, Kini Tampil dengan Konten Vertikal Ala TikTok bersumber dari liputan6  
<https://www.liputan6.com/tekno/read/5229410/spotify-ungkap-desain-baru-kini-tampil-dengan-konten-vertikal-ala-tiktok>
  - Siap-siap, Spotify Bakal Punya Tampilan Baru Mirip TikTok bersumber dari detik.com <https://inet.detik.com/mobile-apps/d-6611297/siap-siap-spotify-bakal-punya-tampilan-baru-mirip-tiktok>

Dari 2 sumber berita di atas, *berita1.txt* menyimpan isi berita sumber liputan6 dan *berita2.txt* menyimpan isi berita sumber detik.com

### 2. Kita buat modul dengan script di bawah ini

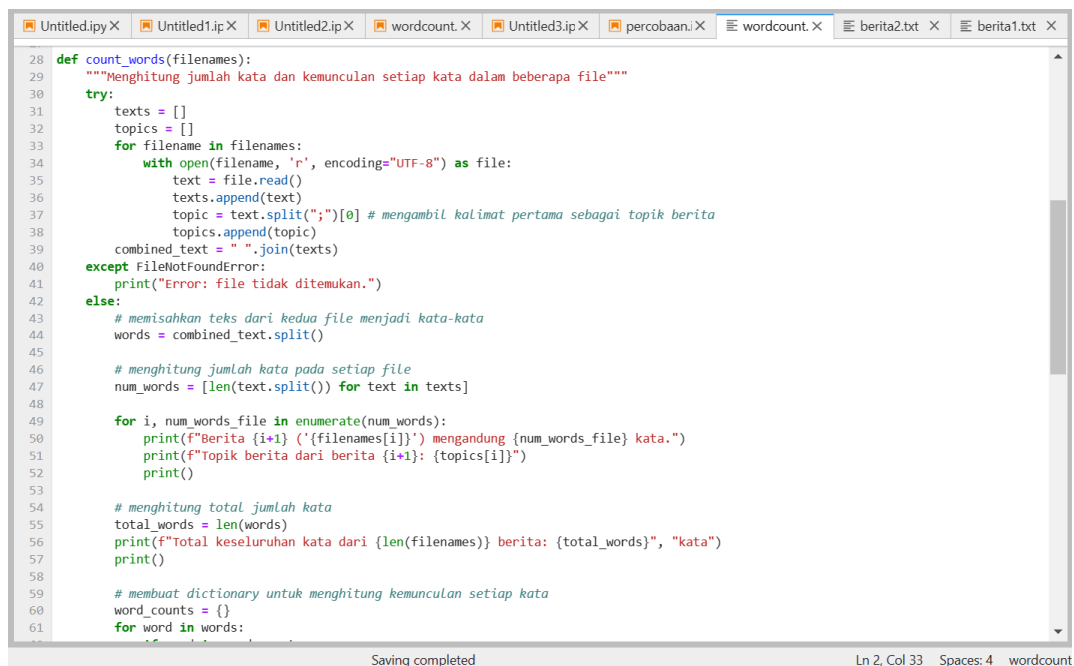


```
1 from collections import defaultdict
2 from multiprocessing import Pool
3 import string
4
5 def mapper(filename):
6     """Membuat pasangan key-value untuk setiap kata pada teks dalam file"""
7
8     with open(filename, 'r', encoding='utf-8') as file:
9         text = file.read()
10
11     translator = str.maketrans('', '', string.punctuation)
12     text = text.lower().translate(translator)
13
14     words = text.split()
15     pairs = [(word, 1) for word in words]
16
17     return pairs
18
19 def reducer(pairs):
20     """Menghitung jumlah kemunculan kata pada teks"""
21
22     counts = defaultdict(int)
23     for pair in pairs:
24         counts[pair[0]] += pair[1]
25
26     return counts
27
28 def count_words(filename):
29     """Menghitung jumlah kata dan kemunculan setiap kata dalam beberapa file"""
30     try:
31         texts = []
32         topics = []
33         for filename in filenames:
34             with open(filename, 'r', encoding="UTF-8") as file:
```

Keterangan script:

- Import modul modul yang diperlukan untuk program MapReduce WordCount yakni:
  - *defaultdict* dari library *collections* untuk membuat sebuah objek dictionary yang akan digunakan dalam proses reduksi di program MapReduce.
  - Modul *string* untuk membersihkan teks dari tanda baca dan mengubah semua karakter menjadi huruf kecil sebelum kata-kata dipisahkan dan menghasilkan pasangan key-value untuk setiap kata dalam teks yang siap digunakan pada proses MapReduce
  - Modul class *Pool* dari library *collections* untuk proses fungsi pemetaan (mapper) pada satu file input dan mengembalikan hasil dalam bentuk list pasangan key-value yang akan dikumpulkan dan digabungkan dalam proses reduksi (reducer) sehingga menghasilkan output akhir.

- Buat beberapa fungsi dengan *def*
  - *Mapper* digunakan untuk melakukan pemetaan pada setiap file input dan menghasilkan pasangan key-value untuk setiap kata dalam teks di file berita.  
Prosesnya yakni pertama-tama teks akan dibaca dalam file berita menggunakan fungsi *open* dan *read*. Selanjutnya tanda baca yang ada dalam teks akan dihilangkan dengan fungsi *str.maketrans* dan *translate* serta mengubah seluruh karakter dalam teks menjadi huruf kecil dengan fungsi *lower*. Setelah itu, menggunakan fungsi *split* teks akan dipecah menjadi kata per kata dan untuk setiap kata dibuat sebagai pasangan key-value dimana key berupa kata tersebut dan value berupa integer 1. Dan pasangan key-value tersebut dikumpulkan dalam list dan dikembalikan sebagai output dari fungsi *mapper*.
  - *Reducer* digunakan untuk melakukan reduksi pada pasangan key-value yang dihasilkan dalam proses pemetaan di setiap inputan file berita  
Prosesnya yakni list yang berisi pasangan key-value dari mapper akan diterima sebagai inputan yang akan digabungkan dan dihitung jumlah kemunculan setiap kata pada teks dengan objek *defaultdict(int)* sebagai penyimpanan. Selanjutnya menambahkan value dari key pada objek dengan operator *+=* sehingga objek akan terus memperbarui dan menghasilkan jumlah kemunculan kata yang akurat dari teks berita lalu hasilnya akan dikembalikan sebagai output dari fungsi *reducer*.



```

28 def count_words(filenamees):
29     """Menghitung jumlah kata dan kemunculan setiap kata dalam beberapa file"""
30     try:
31         texts = []
32         topics = []
33         for filename in filenamees:
34             with open(filename, 'r', encoding="UTF-8") as file:
35                 text = file.read()
36                 texts.append(text)
37                 topic = text.split(";")[0] # mengambil kalimat pertama sebagai topik berita
38                 topics.append(topic)
39             combined_text = " ".join(texts)
40     except FileNotFoundError:
41         print("Error: file tidak ditemukan.")
42     else:
43         # memisahkan teks dari kedua file menjadi kata-kata
44         words = combined_text.split()
45
46         # menghitung jumlah kata pada setiap file
47         num_words = [len(text.split()) for text in texts]
48
49         for i, num_words_file in enumerate(num_words):
50             print(f"Berita {i+1} ({filenamees[i]}) mengandung {num_words_file} kata.")
51             print(f"Topik berita dari berita {i+1}: {topics[i]}")
52             print()
53
54         # menghitung total jumlah kata
55         total_words = len(words)
56         print(f"Total keseluruhan kata dari {len(filenamees)} berita: {total_words}", "kata")
57         print()
58
59         # membuat dictionary untuk menghitung kemunculan setiap kata
60         word_counts = {}
61         for word in words:

```

```
44 words = combined_text.split()
45
46 # menghitung jumlah kata pada setiap file
47 num_words = [len(text.split()) for text in texts]
48
49 for i, num_words_file in enumerate(num_words):
50     print(f"Berita {i+1} ('{filenames[i]}') mengandung {num_words_file} kata.")
51     print(f"Topik berita dari berita {i+1}: {topics[i]}")
52     print()
53
54 # menghitung total jumlah kata
55 total_words = len(words)
56 print(f"Total keseluruhan kata dari {len(filenames)} berita: {total_words}, "kata")
57 print()
58
59 # membuat dictionary untuk menghitung kemunculan setiap kata
60 word_counts = {}
61 for word in words:
62     if word in word_counts:
63         word_counts[word] += 1
64     else:
65         word_counts[word] = 1
66
67 # menampilkan jumlah kemunculan setiap kata, diurutkan dari abjad
68 print("Jumlah kemunculan setiap kata:")
69 for word, count in sorted(word_counts.items()):
70     print(f"{word}: {count}")
71
72 return {
73     'total_words': total_words,
74     'word_counts': word_counts,
75     'num_words': num_words,
76     'topics': topics,
77 }
```

- *Count\_words* untuk menghitung jumlah kata dan kemunculan setiap kata pada beberapa file dengan memberikan argumen *filenames* yakni daftar nama file berita yang akan dihitung jumlah katanya.

Prosesnya yakni fungsi akan membaca setiap file berita dan menyimpannya dalam daftar *texts* juga mengambil kalimat pertama dari file berita sebagai topik berita dengan menyimpannya dalam *topics*. Selanjutnya semua teks dalam daftar *texts* akan digabungkan menjadi satu teks dan disimpan dalam variabel *combined\_text* lalu akan dipecah menjadi kata per kata dan dihitung jumlah katanya.

Fungsi akan membuat dictionary berisi kemunculan setiap kata pada teks yang telah digabungkan dimana tiap kata akan dicek apakah sudah tersedia dalam dictionary, jika sudah maka nilai pada dictionary akan ditambah 1 sedangkan jika belum maka kata tersebut akan ditambahkan dalam dictionary dengan nilai awal 1 sebagai kata baru.

Fungsi juga akan menampilkan jumlah kemunculan setiap kata yang diurutkan sesuai abjad mulai dari tanda baca, huruf berawalan kapital, dan huruf berawalan kecil. Selain itu, fungsi juga mengembalikan dictionary yang berisi informasi tentang total jumlah kata, jumlah kemunculan setiap kata, jumlah kata pada setiap file berita, dan topik berita dari setiap file berita.

- Setelah modul dibuat maka modul disimpan dengan nama *WordCount.py* siap di running kemudian buat lembar ipynb baru untuk mengimport modul dan menghasilkan output.
- Pada lembar ipynb baru, import modul *WordCount* yang sebelumnya telah dibuat. Kemudian deklarasikan variabel *result* untuk memanggil fungsi *count\_words* dan diberikan nama file berita txt (berita1.txt dan berita2.txt) kemudian di running dan akan menghasilkan output berupa Jumlah total kata dari kedua file berita, dictionary dari jumlah kemunculan setiap kata dari kedua file berita, list dari jumlah kata setiap file berita, dan list dari topik setiap file berita

```
[2]: import wordcount

# menghitung jumlah kata dan frekuensi kemunculan kata pada file1.txt dan file2.txt
result = wordcount.count_words(['berita1.txt', 'berita2.txt'])

Berita 1 ('berita1.txt') mengandung 534 kata.
Topik berita dari berita 1: Spotify Ungkap Desain Baru, Kini Tampil dengan Konten Vertikal Ala TikTok

Berita 2 ('berita2.txt') mengandung 346 kata.
Topik berita dari berita 2: Siap-siap, Spotify Bakal Punya Tampilan Baru Mirip TikTok

Total keseluruhan kata dari 2 berita: 880 kata

Jumlah kemunculan setiap kata:
"Music": 1
&: 2
'+': 1
'Music': 1
'Podcasts': 1
(11/3/2023),: 1
(9/3/2023).: 1
-: 2
2: 1
2023: 1
2023,: 1
3: 1
4: 4
50: 1
500: 1
70.000: 1
AI: 4
Ala: 1
Anchor: 1
Ek: 2
Enhance: 1
Global: 1
Head: 1
Homescreen: 1
Instagram: 1
Instagram,: 1
Instagram.: 2
Jakarta: 2
Jauh: 1
Jenis: 1
Jika: 1
Julie: 1
Kamis: 1
Kehadiran: 1
Kembangkan: 1
Kini: 1
Kini,: 1
Konten: 2
Kontradiksi: 1
Lebih: 1
Lewat: 1
Liputan6.com,: 1
McNamara: 1
McNamara,: 1
Melalui: 1
Menurut: 3
Menurutnya: 1
Meskipun: 1
Minggu: 1
Mirip: 1
Namun: 3
Namun,: 1
Namun,: 1
On: 2
On.: 1
Pada: 1
Pendiri: 1
Pengguna: 1
Perubahan: 1
Podcast: 2
Premium.: 1
Preview: 1
Punya: 1
Reels.: 1
Saat: 1
Sama: 1
Sebelumnya,: 1
Sedangkan: 1
Selain: 2
Selama: 1
Sesuai: 1
Shows': 1
Shows": 1
Shuffle: 3
Shuffle.: 1
Siap-siap,: 1
Smart: 4
Spotify: 29
Spotify,: 2
Spotify.: 3
Stream: 3
Studios,: 1
Tak: 1
Tampil: 1
Tampilan: 3
```

Berikut adalah hasil WordCount tiap kata:

Jumlah kemunculan setiap kata:	Ek: 2	
"Music": 1	Enhance: 1	Namun,: 1
&: 2	Global: 1	On: 2
': 1	Head: 1	On.: 1
'Music': 1	Homescreen: 1	Pada: 1
'Podcasts': 1	Instagram: 1	Pendiri: 1
(11/3/2023),: 1	Instagram,: 1	Pengguna: 1
(9/3/2023).: 1	Instagram.: 2	Perubahan: 1
-: 2	Jakarta: 2	Podcast: 2
2: 1	Jauh: 1	Premium.: 1
2023: 1	Jenis: 1	Preview: 1
2023,: 1	Jika: 1	Punya: 1
3: 1	Julie: 1	Reels.: 1
4: 4	Kamis: 1	Saat: 1
50: 1	Kehadiran: 1	Sama: 1
500: 1	Kembangkan: 1	Sebelumnya,: 1
70.000: 1	Kini: 1	Sedangkan: 1
AI: 4	Kini,: 1	Selain: 2
Ala: 1	Konten: 2	Selama: 1
Anchor.: 1	Kontradiksi: 1	Sesuai: 1
Bakal: 1	Lebih: 1	Shows': 1
Baru: 2	Lewat: 1	Shows": 1
Baru,: 1	Liputan6.com,: 1	Shuffle: 3
Beragam: 1	McNamara: 1	Shuffle.: 1
CEO: 2	McNamara,: 1	Siap-siap,: 1
Canvas: 1	Melalui: 1	Smart: 4
DJ: 2	Menurut: 3	Spotify: 29
Dalam: 2	Menurutnya: 1	Spotify,: 2
Daniel: 2	Meskipun: 1	Spotify.: 3
Dengan: 2	Minggu: 1	Stream: 3
Desain: 2	Mirip: 1	Studios,: 1
Di: 1	Namun: 3	Tak: 1
Ek: 2	Namun,: 1	Tampil: 1
		Tampilan: 3

Tampilan: 3	barunya: 2
Tampilkan: 1	bawah: 2
The: 3	beberapa: 2
Tidak: 1	belum: 1
TikTok: 2	bentuk: 2
TikTok.: 2	berada: 1
TikTok;: 2	beragam.: 1
URL.: 2	beralih: 1
Ungkap: 1	berbagai: 5
Untuk: 2	berdampingan: 1
Update: 1	berharap: 1
Verge,: 2	berlanjut.: 1
Verge.: 1	berlebihan: 1
Vertikal: 1	bertahun-tahun,: 1
Video: 2	berukuran: 1
YouTube,: 1	berupaya: 2
acara: 1	besar: 2
ada,: 1	bisa: 7
adalah: 1	buku: 1
agar: 2	cara: 1
akan: 21	cepat,: 1
akses.: 1	cocok: 1
ala: 1	content: 1
alami: 1	cover: 1
album: 4	creator.: 1
antara: 2	daftar: 1
antarmuka: 1	dalam: 6
aplikasi: 7	dan: 21
aplikasinya: 1	dapat: 7
artis: 1	dari: 12
atas: 2	datang,: 1
atau: 4	dengan: 9
audio: 1	depan.: 1
audio,: 1	desain: 7
audiobook: 1	di: 14
bagi: 2	di-scroll: 1
bagian: 3	dibawa: 2
banyak: 7	didengarkan.: 1
baru: 14	digulir: 1
baru,: 1	digulirkan: 2
barunya,: 2	diharapkan: 1
diharapkan: 1	
dikenal: 1	
dikutip: 2	
dilakukan,: 1	
dilansir: 1	
dimaksudkan: 1	
dipersonalisasi: 1	
dipilih: 1	
diputar: 1	
diputar,: 1	
direkomendasikan.: 1	
dirombak: 1	
disambut: 2	
disediakan: 1	
disukai.: 1	
ditampilkan: 2	
ditonton: 1	
dulu: 1	
dunia: 1	
event: 1	
feed: 3	
fitur: 7	
gambar-gambar: 1	
gelaran: 1	
grid: 1	
hal: 1	
hal-hal: 2	
halaman: 7	
hanya: 7	
hidup: 1	
hingga: 1	
home: 2	
homescreen.: 1	
<a href="https://inet.detik.com/mobile-apps/d-6611297/siap-siap-spotify-bakal-punya-tampilan-baru-mirip-tiktok">https://inet.detik.com/mobile-apps/d-6611297/siap-siap-spotify-bakal-punya-tampilan-baru-mirip-tiktok</a> : 1	
<a href="https://www.liputan6.com/tekno/read/5229410/spotify-ungkap-desain-baru-kini-tampil-dengan-konten-vertikal-ala-tiktok">https://www.liputan6.com/tekno/read/5229410/spotify-ungkap-desain-baru-kini-tampil-dengan-konten-vertikal-ala-tiktok</a> : 1	
hubungan: 1	
ia: 3	
ingin: 3	
ini: 18	
ini,: 3	
ini.: 2	

ini,: 3	memanfaatkan: 1	menyebutkan: 1	playlist.: 1	
ini.: 2	membawa: 1	menyelipkan: 1	podcast: 6	
interaktif.: 1	membedakan,: 1	mereka: 4	podcast,: 4	
itu: 1	memberikan: 2	mereka.: 1	podcast.: 1	
itu,: 4	membuat: 2	merekomendasikan: 1	podcaster: 1	tampilan: 6
jadi: 1	membuka: 1	merilis: 1	preview: 2	telah: 4
jenis: 1	memilih: 2	merupakan: 2	preview.: 1	tentu: 1
juga: 8	memiliki: 2	mirip: 3	putar: 1	terbaru: 1
jumlahnya: 1	mempengaruhi: 1	muda.: 1	radio.: 1	terbaru: 1
juta: 2	memperkenalkan: 1	muncul: 1	rekomendasi: 3	terbesar: 1
kami: 1	memperlihatkan: 1	mungkin: 4	rekomendasi,: 1	terbiasa: 1
ke: 6	mempermudah: 1	musik: 4	saat: 1	terdengar: 1
kebiasaan: 1	memutar: 1	musik,: 3	saja: 2	terlihat: 1
kemungkinan: 1	menambahkannya: 1	musik.: 3	salah: 3	tersebut: 1
kepada: 1	menampilkan: 2	namanya,: 1	satu: 3	tersebut,: 1
keterangannya: 1	menarik: 2	of: 1	scroll: 1	tersedia: 2
ketika: 1	mencakup: 1	opsi: 1	sebagai: 4	terus: 2
khusus: 2	mendaftar: 1	otomatis: 1	sebagian: 2	tetapi: 1
klik: 1	mendapatkan: 1	pada: 3	sebelumnya: 1	tombol: 1
klip: 2	mendengarkan: 2	paling: 1	sebelumnya,: 1	turut: 1
konten: 13	menelusuri: 1	pekan: 1	sebuah: 1	ujarnya,: 1
konten.: 1	menempatkan: 1	pelanggan: 1	secara: 5	untuk: 13
kotak-kotak: 1	menemukan: 3	pembaruan: 2	sedang: 1	update: 2
kreator: 1	menerbitkan: 1	pembuat: 1	sedikit.: 1	usia: 1
kualitasnya: 1	mengakses: 1	pemutar: 1	segera: 1	utama: 1
lagi: 1	mengatakan: 1	pendengar: 1	sehingga: 2	versi: 1
lagu: 2	menghadirkan: 1	penggemar.: 1	sekadar: 1	vertikal: 3
lain,: 1	menghadirkannya: 1	pengguna: 18	selama: 1	vertikal,: 2
lainnya: 2	mengharapkan: 1	pengguna,: 2	seluruh: 1	video: 13
lainnya,: 1	menghiasi: 1	pengguna.: 1	sepenuhnya.: 1	visual: 1
lainnya.: 1	mengikuti: 1	penuh: 1	seperti: 6	wadah: 1
lalu.: 1	mengingatkan: 1	penuh,: 1	sering: 1	waktu: 1
lanjut.: 1	mengklik: 1	penyiar: 1	sesuai: 1	yaitu: 1
layaknya: 1	mengubah: 1	perkembangannya: 1	siapa: 1	yang: 31
layar: 1	mengumumkan: 1	pertumbuhan: 1	sisi: 1	“Podcast: 1
lebih: 11	mengungkap: 1	perubahan: 4	streaming: 2	“Saat: 1
masih: 2	meningkatkan: 1	perusahaan,: 1	suara: 1	
melebarkan: 1	menjadi: 2	pilihan: 2	sudah: 2	
melihat: 3	menjelajahi: 2	platform: 7	sukai: 1	
meluncurkan: 1	menuturkan,: 1	playlist: 5	tab: 1	
memanfaatkan: 1	menyebutkan: 1	playlist.: 1	tampilan: 6	



Berikut kode lengkapnya untuk modul:

```
from collections import defaultdict
```

```
from multiprocessing import Pool
```

```
import string
```

```
def mapper(filename):
```

```
    """Membuat pasangan key-value untuk setiap kata pada teks dalam file"""
```

```
    with open(filename, 'r', encoding='utf-8') as file:
```

```
        text = file.read()
```

```
    translator = str.maketrans("", "", string.punctuation)
```

```
    text = text.lower().translate(translator)
```

```
    words = text.split()
```

```
    pairs = [(word, 1) for word in words]
```

```
    return pairs
```

```
def reducer(pairs):
```

```
    """Menghitung jumlah kemunculan kata pada teks"""
```

```
counts = defaultdict(int)

for pair in pairs:

    counts[pair[0]] += pair[1]
```

```
return counts
```

```
def count_words(filename):
```

```
    """Menghitung jumlah kata dan kemunculan setiap kata dalam beberapa
file"""
```

```
    try:
```

```
        texts = []
```

```
        topics = []
```

```
        for filename in filenames:
```

```
            with open(filename, 'r', encoding="UTF-8") as file:
```

```
                text = file.read()
```

```
                texts.append(text)
```

```
                topic = text.split(";")[0] # mengambil kalimat pertama sebagai topik
berita
```

```
                topics.append(topic)
```

```
        combined_text = " ".join(texts)
```

```
    except FileNotFoundError:
```

```
        print("Error: file tidak ditemukan.")
```

```
    else:
```

```
        # memisahkan teks dari kedua file menjadi kata-kata
```

```

words = combined_text.split()

# menghitung jumlah kata pada setiap file
num_words = [len(text.split()) for text in texts]

for i, num_words_file in enumerate(num_words):
    print(f"Berita {i+1} ({filenames[i]}) mengandung {num_words_file} kata.")
    print(f"Topik berita dari berita {i+1}: {topics[i]}")
    print()

# menghitung total jumlah kata
total_words = len(words)

print(f"Total keseluruhan kata dari {len(filenames)} berita: {total_words}", "kata")

print()

# membuat dictionary untuk menghitung kemunculan setiap kata
word_counts = {}

for word in words:
    if word in word_counts:
        word_counts[word] += 1
    else:
        word_counts[word] = 1

```

```
# menampilkan jumlah kemunculan setiap kata, diurutkan dari abjad

print("Jumlah kemunculan setiap kata:")

for word, count in sorted(word_counts.items()):

    print(f"{word}: {count}")


return {

    'total_words': total_words,

    'word_counts': word_counts,

    'num_words': num_words,

    'topics': topics,

}
```