



Universitas Indonesia

HADOOP VS JAVA WORDCOUNT



Oleh :
Kelompok 8



KELOMPOK 8

- Mohammad varrel bramasta - 2106733811
- Eriqo Arief Wicaksono - 2106638406
- Michael Gunawan - 2106731195
- M Farrel Athaillah Nugroho 2106733875

CODE JAVA

JAVA CODE

Code disamping adalah salah satu implementasi wordcount untuk file berukuran 1GB. Adapun cara untuk mengganti file.txt dengan ukuran size yang berbeda adalah dengan cara merubah filepath pada line ke-9

```
1 import java.io.BufferedReader;
2 import java.io.FileReader;
3 import java.util.HashMap;
4 import java.util.Map;
5 public class wordcount_1GB {
    Run | Debug
6     public static void main(String args[]) throws Exception {
7
8         System.out.println("Counting Words");
9         FileReader fr = new FileReader(fileName:"gutenberg-1G.txt"); //membaca file
10        BufferedReader br = new BufferedReader(fr); //deklarasi br sebagai buffer reader
11        String line = br.readLine();
12        Map<String, Integer> wordCountMap = new HashMap<>();
13
14        while (line != null) {
15            String[] words = line.split(regex:"\\s+");
16            for (String word : words) {
17                wordCountMap.put(word, wordCountMap.getOrDefault(word, defaultValue:0) + 1);
18            }
19            line = br.readLine();
20        }
21
22        for (Map.Entry<String, Integer> entry : wordCountMap.entrySet()) {
23            System.out.println(entry.getKey() + ", Count: " + entry.getValue());
24        }
25        br.close();
26        fr.close();
27    }
28 }
```




REFRENSI CODE JAVA



**[https://stackoverflow.com/questions/47594679
/how-to-count-words-in-a-text-file-java-8-style](https://stackoverflow.com/questions/47594679/how-to-count-words-in-a-text-file-java-8-style)**





HARDWARE SPESIFIKASI 1

CPU - AMD Ryzen 7 4800H

Base speed:	2,90 GHz
Sockets:	1
Cores:	8
Logical processors:	16
Virtualization:	Enabled
L1 cache:	512 KB
L2 cache:	4,0 MB
L3 cache:	8,0 MB

RAM - 16GB

Speed:	3200 MHz
Slots used:	2 of 2
Form factor:	SODIMM
Hardware reserved:	653 MB



RESULTS 1

RESULT1 - JAVA



100 MB:

```
ptime 1.0 for Win32, Freeware - http://www.pc-tools.net/  
Copyright(C) 2002, Jem Berkes <jberkes@pc-tools.net>  
  
=== java WordCount ===  
Counting Words  
  
Execution time: 7.094 s
```

200 MB:

```
ptime 1.0 for Win32, Freeware - http://www.pc-tools.net/  
Copyright(C) 2002, Jem Berkes <jberkes@pc-tools.net>  
  
=== java WordCount ===  
Counting Words  
  
Execution time: 13.447 s
```


RESULT1 - JAVA



500 MB:

```
ptime 1.0 for Win32, Freeware - http://www.pc-tools.net/  
Copyright(C) 2002, Jem Berkes <jberkes@pc-tools.net>  
  
=== java WordCount ===  
Counting Words  
  
Execution time: 34.311 s
```

1 GB:

```
ptime 1.0 for Win32, Freeware - http://www.pc-tools.net/  
Copyright(C) 2002, Jem Berkes <jberkes@pc-tools.net>  
  
=== java WordCount ===  
Counting Words  
  
Execution time: 71.439 s
```


RESULT 1 - HADOOP

100 MB:

```
Bytes Read=104857600  
File Output Format Counters  
Bytes Written=8221780  
Execution time: 67.015 s
```

200 MB:

```
File Input Format Counters  
Bytes Read=209719296  
File Output Format Counters  
Bytes Written=13900354  
Execution time: 67.828 s
```


RESULT 1 - HADOOP

500 MB:

```
File Output Format Counters
  Bytes Written=27023894
Execution time: 78.415 s
```

1 GB:

```
File Input Format Counters
  Bytes Read=1073770496
File Output Format Counters
  Bytes Written=45388248
Execution time: 129.123 s
```


TABEL PERBANDINGAN 1Q

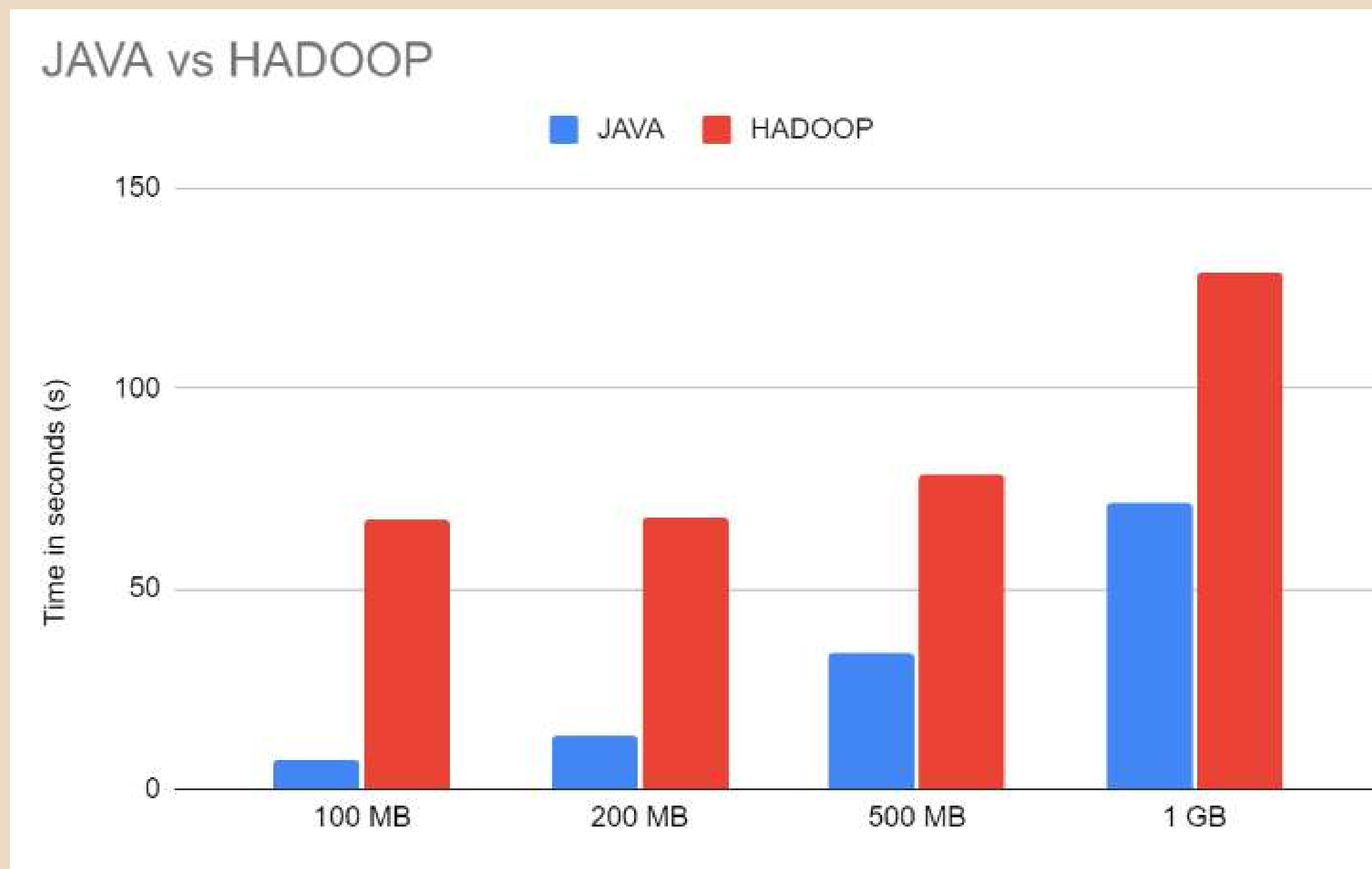
SIZE - TIME

	100 MB	200 MB	500 MB	1 GB
JAVA	7.094 s	13.447 s	34.311 s	71.439 s
HADOOP	67.015 s	67.828 s	78.415 s	129.123 s

GRAFIK 1



TIME



SIZE



HARDWARE SPESIFIKASI 2



Current Date/Time: 20 June 2023, 18:12:43

Computer Name: BRAMASTA

Operating System: Windows 11 Home 64-bit (10.0, Build 22621)

Language: English (Regional Setting: English)

System Manufacturer: LENOVO

System Model: 82B5

BIOS: EUCN31WW

Processor: AMD Ryzen 5 4600H with Radeon Graphics (12 CPUs), ~3.0GHz

Memory: 16384MB RAM

Page file: 11253MB used, 6919MB available

DirectX Version: DirectX 12

Device

Name: NVIDIA GeForce GTX 1650 Ti

Manufacturer: NVIDIA

Chip Type: NVIDIA GeForce GTX 1650 Ti


DAC Type: Integrated RAMDAC

Device Type: Full Display Device

Approx. Total Memory: 11819 MB

Display Memory (VRAM): 3949 MB

Shared Memory: 7870 MB



RESULTS 2

RESULT 2 - JAVA



100MB FILE.TXT :

```
[Done] exited with code=0 in 12.816 seconds
```

200MB FILE.TXT :

```
[Done] exited with code=0 in 25.622 seconds
```

500MB FILE.TXT :

```
[Done] exited with code=0 in 40.558 seconds
```

1GB FILE.TXT :

```
[Done] exited with code=0 in 95.466 seconds
```


RESULT 2 - HADOOP

100MB FILE.TXT :

```
Execution time: 44.505 s
```

200MB FILE.TXT :

```
Execution time: 48.295 s
```

500MB FILE.TXT :

```
Execution time: 79.957 s
```

1GB FILE.TXT :

```
Execution time: 124.420 s
```

TABEL PERBANDINGAN 2Q

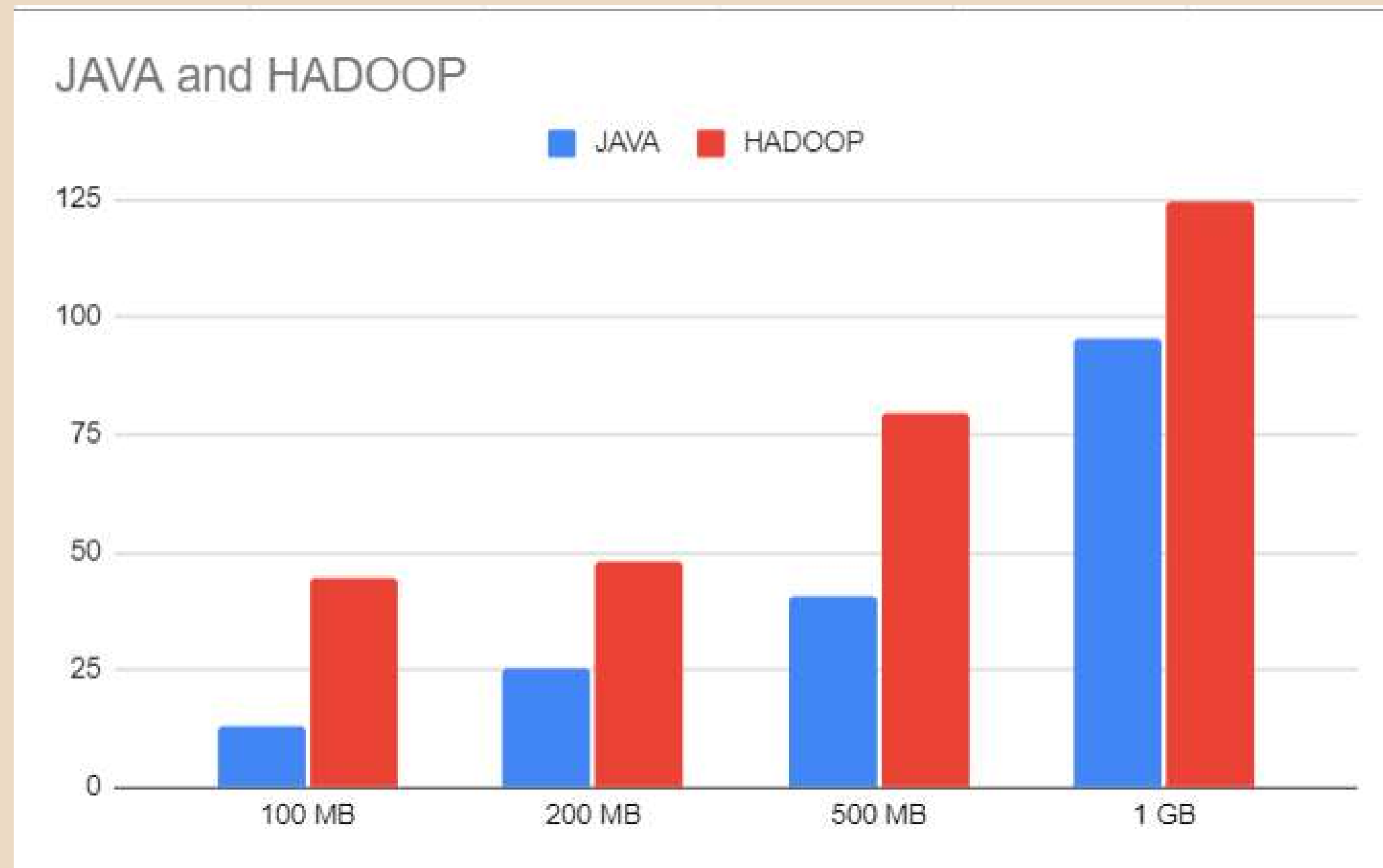
SIZE - TIME

	100 MB	200 MB	500 MB	1 GB
JAVA	12.816s	25.622s	40.558s	95.446s
HADOOP	44.505s	48.295s	79.957s	124.420s

GRAFIK 2



TIME



SIZE



ANALISIS




ANALISIS



Hasil dari percobaan sebelumnya menunjukkan bahwa algoritma wordcount yang diimplementasikan dalam bahasa pemrograman Java memiliki kecepatan eksekusi yang lebih cepat daripada pendekatan yang menggunakan Hadoop, untuk file uji dengan ukuran 100MB, 200MB, 500MB, dan 1GB. Namun, perbedaan yang signifikan antara kedua pendekatan tersebut terletak pada hasil output berkas .txt yang dihasilkan.

File .txt yang dihasilkan oleh program Java tidak mengurutkan kata-kata secara alfabetikal, sedangkan keluaran yang dihasilkan oleh Hadoop sudah diurutkan dengan rapi dan teratur. Faktor inilah yang dinilai mempengaruhi kecepatan eksekusi algoritma wordcount dalam Java, dibandingkan dengan algoritma wordcount dalam Hadoop dengan cukup signifikan.





KESIMPULAN



KESIMPULAN



Dapat disimpulkan bawah bahasa pemrograman Java lebih cepat ketimbang Hadoop dalam aspek kecepatan eksekusi baik untuk kasus file berukuran 100MB, 200MB, 500MB, dan 1GB. Adapun kekurangan yang dimiliki Java adalah tidak bisa mengurutkan kata perkata secara alfabetikal sehingga dinilai mempengaruhi kecepatan waktu eksekusi dari adanya sorting secara alfabet tadi..





Universitas Indonesia

THANK
YOU



Oleh :
Kelompok 8

