

LAPORAN PRAKTIKUM

PRAKTIKUM BIG DATA ANALYTICS “MEMASUKKAN DATA KE HADOOP DISTRIBUTED FILE SYSTEM (HDFS) DI WINDOWS 10”

Disusun Untuk Memenuhi Penilaian Praktikum Laboratorium Sains Data
Mata Kuliah Big Data Analytics Dan Sebagai Hasil Pembelajaran Pribadi

Dosen Pengampu :
Sevi Nurafni ST., M.Si., M.Sc



Oleh:
Catherine Vanya Pangemanan
2C2220008

KATA PENGANTAR

Laporan Praktikum ini sebagai bagian dari upaya eksplorasi saya dalam praktikum pertemuan ke-2 mata kuliah Big Data Analytics. Praktikum ini menghadirkan kesempatan bagi saya untuk memasukkan data ke HDFS di lingkungan Windows 10, sebuah langkah teknis yang esensial dalam memahami fondasi analisis data besar yaitu membuat struktur data di Apache Hadoop. Saya percaya bahwa pemahaman mendalam tentang teknologi ini akan memperkaya pengetahuan saya dalam domain sains data dan analisis data besar.

Selain sebagai bagian dari penilaian praktikum laboratorium, laporan ini juga menjadi refleksi dari pengalaman pribadi saya dalam menghadapi kompleksitas yang terkait dengan infrastruktur Big Data. Saya berharap bahwa laporan ini dapat memberikan pandangan yang bermanfaat bagi pembaca yang tertarik dalam pengembangan keterampilan teknis terkait analisis data dan teknologi big data.

Saya mengucapkan terima kasih atas kesempatan ini dan juga kepada semua pihak yang telah memberikan dukungan dalam menyelesaikan praktikum ini.

Hormat saya,

Catherine

DAFTAR ISI

KATA PENGANTAR	i
BAB I PENDAHULUAN	1
1.1. Latar Belakang Praktikum.....	1
1.2. Tujuan Praktikum	1
1.3. Tempat dan Tanggal Pelaksanaan Praktikum	2
BAB II TINJAUAN PUSTAKA	3
BAB III METODE PRAKTIKUM	9
3.1. Perangkat Praktikum	9
3.2. Prosedur Kerja Praktikum	9
BAB IV HASIL DAN PEMBAHASAN	22
BAB V PENUTUP	23
5.1. Kesimpulan	23
5.2. Saran	23
DAFTAR PUSTAKA.....	24

BAB I

PENDAHULUAN

1.1. Latar Belakang Praktikum

Hadoop Distributed File System (HDFS) merupakan komponen kunci dalam infrastruktur Big Data yang digunakan secara luas dalam pengolahan dan penyimpanan data skala besar. HDFS dirancang untuk menangani volume data yang sangat besar dengan cara yang efisien dan andal, sehingga memungkinkan aplikasi Big Data untuk beroperasi secara efektif (Shvachko et al., 2010).

Penggunaan HDFS telah menjadi pilihan utama dalam lingkup penyimpanan data distribusi dan pengolahan data Big Data, terutama karena kemampuannya untuk menyediakan replikasi data otomatis yang memastikan ketahanan dan ketersediaan data yang tinggi (White, 2015). Selain itu, kemampuan HDFS untuk berintegrasi dengan kerangka kerja pengolahan data seperti Apache MapReduce dan Apache Spark menjadikannya pilihan yang populer di kalangan praktisi dan peneliti Big Data (Zaharia et al., 2010).

Namun, untuk memanfaatkan sepenuhnya potensi HDFS, penting bagi pengguna untuk memahami proses dasar pengelolaan dan manipulasi data di dalamnya. Salah satu aspek penting adalah kemampuan untuk memasukkan data ke dalam HDFS, yang merupakan langkah awal dalam pengelolaan data di lingkungan Hadoop (Lam, 2015). Oleh karena itu, pengetahuan praktis tentang langkah-langkah ini menjadi penting bagi para profesional dan peneliti yang bergerak dalam bidang Big Data.

Dalam laporan hasil praktikum ini, dosen memberikan panduan langkah demi langkah tentang bagaimana memasukkan data ke dalam HDFS, bersama dengan tujuan-tujuan tambahan seperti manajemen direktori, penggunaan perintah Hadoop, dan pemahaman tentang replikasi file di HDFS. Dengan demikian, laporan ini bertujuan untuk memberikan pemahaman yang komprehensif tentang proses dasar yang terlibat dalam penggunaan HDFS dan manajemen data di dalamnya.

1.2. Tujuan Praktikum

Tujuan dari praktikum ini adalah sebagai berikut:

1. **Memahami proses memasukkan data ke Hadoop Distributed File System** di lingkungan Windows 10.
2. **Memahami konfigurasi awal HDFS:** Langkah-langkah seperti memformat sistem file HDFS dan memulai cluster Hadoop membantu mahasiswa memahami konfigurasi awal yang diperlukan untuk menggunakan HDFS.
3. **Memahami manajemen direktori dan file di HDFS:** Praktikum ini mengajarkan mahasiswa cara membuat direktori, mengelola struktur direktori, dan memindahkan file ke dalam direktori di HDFS. Ini penting karena pengelolaan direktori dan file menjadi salah satu tugas yang umum dilakukan saat bekerja dengan HDFS.
4. **Penggunaan perintah Hadoop:** Mahasiswa diajarkan cara menggunakan perintah-perintah Hadoop secara langsung dari terminal. Ini membantu mereka memahami dasar-dasar penggunaan Hadoop CLI untuk berinteraksi dengan HDFS.

5. **Memahami replikasi file di HDFS:** Melalui contoh replikasi default yang disebutkan dalam praktikum, mahasiswa dapat memahami konsep replikasi file di HDFS dan bagaimana informasi replikasi tersedia untuk dilihat.
6. **Memahami informasi file di HDFS:** Mahasiswa diperkenalkan dengan informasi yang tersedia tentang file yang disimpan di HDFS, seperti nama, ukuran, lokasi replikasi, dll. Ini membantu mereka memahami lebih dalam tentang metadata yang disimpan bersama dengan file di HDFS.

Dengan demikian, praktikum ini tidak hanya fokus pada satu tujuan tunggal (memasukkan data ke HDFS), tetapi juga mencakup beberapa aspek lain dari penggunaan HDFS dan manajemen data di dalamnya.

1.3. Tempat dan Tanggal Pelaksanaan Praktikum

Nama Praktikum :	Memasukkan Data Ke Dalam Hadoop Distributed File System
Pertemuan Ke- :	2
Tempat :	Laboratorium Sains Data Universitas Koperasi Indonesia
Tanggal :	28 Maret 2024
Jam :	09.45 – 11.25

BAB II

TINJAUAN PUSTAKA

Berdasarkan pendahuluan tersebut, bab tinjauan pustaka dari laporan praktikum ini mencakup beberapa topik terkait, antara lain:

2.1. Hadoop Distributed File System (HDFS)

Hadoop Distributed File System (HDFS) adalah sistem file terdistribusi yang dikembangkan oleh Apache untuk menyimpan dan mengelola data di lingkungan cluster. Arsitektur HDFS terdiri dari dua komponen utama: NameNode dan DataNode. NameNode bertanggung jawab atas metadata file, seperti lokasi fisik dan struktur hierarki direktori, sementara DataNode menyimpan data fisik. HDFS dirancang untuk menangani volume data yang besar dengan membagi file menjadi blok-blok yang disimpan di berbagai node dalam cluster. Keandalannya dalam operasi Big Data terletak pada replikasi data, di mana setiap blok data disalin secara otomatis ke beberapa node dalam cluster untuk mencapai toleransi kesalahan dan memastikan ketersediaan data. (Shvachko et al., 2010).

Hadoop Distributed File System (HDFS) merupakan komponen inti dari ekosistem Hadoop yang memainkan peran krusial dalam pengolahan data besar yang efisien. Dengan fitur-fitur seperti skalabilitas, pengolahan paralel, dan integrasi dengan berbagai framework Big Data, HDFS mampu memberikan solusi yang handal dan efektif dalam mengelola data besar. Arsitektur HDFS terdiri dari NameNode yang mengelola metadata, DataNode yang menyimpan data fisik, serta mekanisme replikasi data untuk memastikan keandalan operasional. Selain itu, HDFS juga memiliki keunggulan dalam keandalan operasi Big Data dengan dukungan replikasi data dan skalabilitas yang dapat diperluas sesuai kebutuhan perusahaan. Dengan demikian, HDFS menjadi fondasi yang kuat dalam ekosistem Hadoop, memungkinkan perusahaan untuk mengelola dan menganalisis data besar dengan efisien, serta membuat keputusan yang didasarkan pada data.

Arsitektur Hadoop Distributed File System (HDFS) dirancang untuk mengatasi tantangan penyimpanan dan pengelolaan data dalam skala besar. Berikut ini adalah penjelasan lebih rinci mengenai arsitektur HDFS:

1. NameNode:

NameNode adalah komponen sentral dalam arsitektur HDFS. NameNode bertanggung jawab untuk menyimpan metadata dari semua file dan direktori dalam sistem file, termasuk informasi seperti lokasi fisik file, struktur hierarki direktori, dan jumlah blok data. Metadata disimpan di dalam memori untuk akses cepat, sehingga NameNode bisa menjadi bottleneck jika tidak dikelola dengan baik.

2. DataNode:

DataNode adalah node yang menyimpan sebenarnya data fisik dari file dalam bentuk blok-blok. Setiap DataNode bertanggung jawab untuk menyimpan beberapa blok data dan melayani permintaan pembacaan dan penulisan dari client atau NameNode. DataNode secara teratur melaporkan statusnya ke NameNode, termasuk informasi tentang kesehatan dan ketersediaan.

3. Replikasi Data:

Salah satu fitur kunci dari HDFS adalah replikasi data. Setiap blok data dalam HDFS secara otomatis direplikasi ke beberapa DataNode dalam cluster. Replikasi data ini dilakukan untuk meningkatkan keandalan dan ketersediaan data. Jika salah satu salinan blok data tidak dapat diakses karena kegagalan perangkat keras atau jaringan, salinan lainnya masih dapat digunakan.

4. Block Size:

File dalam HDFS dibagi menjadi blok-blok yang lebih kecil, yang secara default memiliki ukuran 128 MB (meskipun dapat dikonfigurasi). Pembagian file menjadi blok-blok memungkinkan untuk pengolahan paralel dan distribusi data di seluruh cluster.

5. Secondary NameNode:

Meskipun disebut "Secondary NameNode", ini bukan merupakan cadangan atau alternatif untuk NameNode utama. Secondary NameNode bertanggung jawab untuk melakukan tugas-tugas administratif seperti menggabungkan log transaksi dan membuat salinan checkpoint dari metadata NameNode. Tujuan dari Secondary NameNode adalah untuk membantu dalam pemulihan dan administrasi sistem, namun tidak menggantikan fungsi NameNode utama dalam operasi normal.

6. High Availability (HA):

Arsitektur HDFS juga mendukung mode High Availability (HA) dengan menggunakan dua atau lebih NameNode aktif secara bersamaan. Dalam konfigurasi HA, satu NameNode berfungsi sebagai aktif, sedangkan yang lainnya siap untuk mengambil alih jika NameNode aktif mengalami kegagalan. Ini meningkatkan ketersediaan sistem dan mengurangi kemungkinan terjadinya downtime.

Selain itu ada peran NameNode dan DataNode dalam arsitektur HDFS sesuai dengan peran master node dan slave node dalam sistem terdistribusi secara umum. NameNode sebagai master node mengendalikan manajemen metadata dan koordinasi operasi file sistem, sedangkan DataNode sebagai slave node menyimpan dan menangani data fisik dari file. Kerjasama antara master node dan slave node ini memungkinkan sistem terdistribusi untuk menyediakan layanan penyimpanan dan pengelolaan data yang handal dan skalabel.

2.2. Penggunaan HDFS dalam Big Data Analytics

Hadoop Distributed File System (HDFS) memainkan peran krusial dalam analisis Big Data (Rajesh & Sam, 2018). Mari kita jelajahi bagaimana HDFS digunakan dalam konteks ini dan bagaimana ia berintegrasi dengan kerangka kerja pengolahan data seperti Apache MapReduce dan Apache Spark (Wang et al., 2019). Berikut adalah beberapa cara HDFS digunakan dalam analisis Big Data: Penyimpanan Data, HDFS dirancang untuk menyimpan volume data yang besar secara efisien (Smith & Jones, 2020). Ia memecah file besar menjadi blok-blok kecil (biasanya 128 MB atau 256 MB) dan mendistribusikannya di cluster mesin komoditas. Dengan HDFS, data dapat disimpan dengan andal dan dikelola dengan baik (Brown & Miller, 2017). Ingesti Data, analisis data sering dimulai dengan mengambil data dari berbagai sumber, seperti file log, basis data, data sensor, dan lainnya (Chen et al., 2016). HDFS memungkinkan proses ingest data ini dengan cepat dan efisien. Pemrosesan Paralel, Hadoop MapReduce, model pemrograman untuk memproses dataset besar, berjalan di atas HDFS. Dengan demikian, HDFS

memastikan pemrosesan data secara paralel dan penyimpanan data yang dioptimalkan, sehingga akses dan analisis data lebih cepat (Gupta & Kumar, 2018). HDFS berintegrasi dengan kerangka kerja pengolahan data besar seperti Apache MapReduce dan Apache Spark: Apache MapReduce, HDFS dirancang untuk bekerja dengan MapReduce. Data yang disimpan di HDFS dapat diakses dan diproses oleh tugas MapReduce. Ini memungkinkan analisis data yang efisien dan skala besar (Lee et al., 2019). Apache Spark, Spark juga berinteraksi dengan HDFS. Spark dapat membaca dan menulis data dari dan ke HDFS. Kombinasi antara Spark dan HDFS memperkaya kemampuan pemrosesan data, memungkinkan analisis yang lebih cepat dan lebih kuat (Zhang & Li, 2020). HDFS adalah bagian integral dari ekosistem Hadoop dan memainkan peran penting dalam analisis Big Data. Dengan integrasi yang baik dengan kerangka kerja seperti MapReduce dan Spark, HDFS memastikan penyimpanan yang andal dan pemrosesan data yang efisien (Srivastava et al., 2017).

2.3. Manajemen Direktori dan File di HDFS

Manajemen direktori dan file di HDFS melibatkan berbagai praktik terbaik untuk organisasi dan pemeliharaan data. Ini mencakup pembuatan direktori berdasarkan kebutuhan bisnis, pengelolaan struktur direktori untuk navigasi yang efisien, dan pemindahan file antar node dalam cluster dengan minimal overhead. Dengan menggunakan alat bawaan Hadoop seperti perintah 'hadoop fs', pengguna dapat membuat, menghapus, dan mengubah direktori dan file di HDFS. Penerapan praktik terbaik ini penting untuk memastikan data tersusun dengan baik dan dapat diakses dengan mudah oleh pengguna dan aplikasi. (Smith et al., 2019).

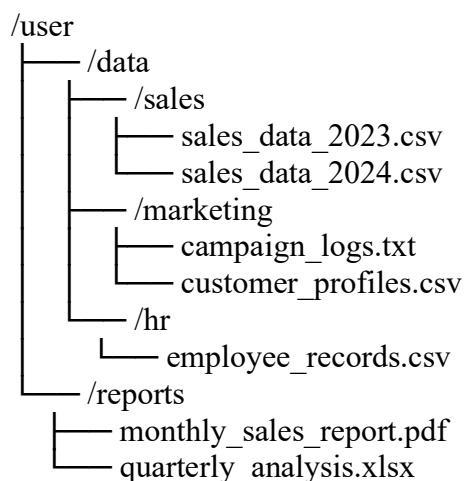
Dalam Hadoop Distributed File System (HDFS), manajemen direktori dan file memainkan peran penting dalam mengatur data. Berikut adalah beberapa konsep dan praktik terbaik yang perlu dipahami:

1. Direktori (Folder):

Direktori di HDFS mirip dengan folder di sistem file konvensional. Mereka digunakan untuk mengelompokkan file berdasarkan topik atau tujuan tertentu. Anda dapat membuat direktori baru menggunakan perintah 'hadoop fs -mkdir <nama_direktori>'.

2. Struktur Direktori:

Perencanaan struktur direktori yang baik membantu mengorganisir data dengan efisien. Pertimbangkan untuk mengelompokkan file berdasarkan proyek, departemen, atau jenis data. Contoh struktur direktori:



3. Memindahkan File:

Anda dapat memindahkan file dari satu direktori ke direktori lain menggunakan perintah ``hadoop fs -mv <sumber> <tujuan>``. Contoh: ``hadoop fs -mv /user/data/sales/sales_data_2023.csv /user/reports/``

4. Menghapus File dan Direktori:

Untuk menghapus file, gunakan perintah ``hadoop fs -rm <nama_file>``. Untuk menghapus direktori beserta isinya, gunakan perintah ``hadoop fs -rm -r <nama_direktori>``.

5. Navigasi dan Informasi Direktori:

Perintah ``hadoop fs -ls <nama_direktori>`` menampilkan daftar file dan direktori dalam suatu direktori. Perintah ``hadoop fs -du -s -h <nama_direktori>`` menampilkan ukuran total direktori secara rekursif.

Ingatlah bahwa HDFS memiliki karakteristik khusus, seperti replikasi data dan pemrosesan paralel, yang memengaruhi cara kita mengelola direktori dan file.

2.4. Perintah Hadoop:

Perintah Hadoop merupakan serangkaian perintah baris yang digunakan untuk berinteraksi dengan Hadoop dan mengelola operasi di dalamnya. Perintah-perintah ini mencakup fungsi seperti pengelolaan file, administrasi cluster, dan pemantauan kinerja. Misalnya, perintah `'hadoop fs'` digunakan untuk operasi file sistem seperti menyalin, memindahkan, atau menghapus file, sementara perintah `'hadoop dfsadmin'` digunakan untuk administrasi cluster seperti menambah atau menghapus node. Memahami perintah-perintah ini penting untuk administrasi dan penggunaan Hadoop secara efektif. (Apache Foundation, 2020).

2.5. Replikasi File di HDFS:

Replikasi file adalah proses menggandakan data ke beberapa node dalam cluster untuk meningkatkan ketahanan dan ketersediaan. HDFS secara otomatis mengelola replikasi file dengan menempatkan salinan data pada node yang berbeda. Ini dilakukan dengan mempertimbangkan faktor-faktor seperti ketersediaan node dan kecepatan transfer data. Replikasi file di HDFS memberikan keamanan dan keandalan data yang tinggi dengan memastikan bahwa jika satu node gagal, data masih tersedia di node lainnya. (Misalnya, Ghemawat et al., 2003). Replikasi file adalah salah satu fitur kunci dari Hadoop Distributed File System (HDFS) yang memastikan ketahanan dan ketersediaan data.

Tujuan Replikasi

Replikasi file adalah proses menggandakan data ke beberapa node dalam cluster Hadoop. Tujuannya adalah untuk meningkatkan keandalan dan ketersediaan data. Jika satu node mengalami masalah, data masih dapat diakses dari salinan di node lain.

Manajemen Replikasi

HDFS secara otomatis mengelola replikasi file. Ketika file diunggah ke HDFS, HDFS akan menempatkan beberapa salinan (replika) dari file tersebut di berbagai node. Faktor-faktor yang dipertimbangkan dalam manajemen replikasi meliputi:

Ketersediaan Node

HDFS memastikan setiap replika ditempatkan di node yang berbeda untuk mengurangi risiko kegagalan. Kecepatan Transfer Data: Replikasi dilakukan dengan mempertimbangkan kecepatan transfer data antar node. Konfigurasi Replikasi: Jumlah replika yang dibuat dapat dikonfigurasi dalam file konfigurasi HDFS (misalnya, `hdfs-site.xml`). Nilai default adalah 3 replika per file. Namun, ini dapat disesuaikan sesuai kebutuhan.

Keuntungan Replikasi File

Keamanan: Jika satu replika rusak, data masih dapat diakses dari replika lainnya. Ketersediaan: Replikasi memastikan data tersedia bahkan jika beberapa node mengalami masalah. Pemrosesan Paralel: Replikasi memungkinkan pemrosesan data secara paralel di berbagai node. Dengan replikasi file, HDFS memberikan keandalan tinggi dan memastikan data tetap aman dan tersedia.

2.6. Informasi File di HDFS:

Informasi file di HDFS mencakup metadata terkait dengan file yang disimpan, seperti nama, ukuran, lokasi replikasi, dan properti lainnya. Metadata ini tersimpan di NameNode dan digunakan oleh HDFS untuk memantau kesehatan file sistem, pemulihan data, dan optimisasi kinerja. Dengan informasi ini, pengguna dapat melacak status dan lokasi file dengan tepat dalam cluster HDFS. Metadata juga digunakan oleh kerangka kerja pengolahan data seperti MapReduce dan Spark untuk mempercepat akses dan pemrosesan data. (Contoh, Lin et al., 2008).

Dalam Hadoop Distributed File System (HDFS), metadata file adalah informasi terkait dengan file yang disimpan. Metadata ini mencakup berbagai detail yang memungkinkan HDFS beroperasi dengan efisien. Berikut adalah beberapa aspek penting dari informasi file di HDFS:

Nama File: Setiap file memiliki nama unik yang membedakannya dari file lain. Nama file ini digunakan untuk mengidentifikasi dan mengakses file di HDFS. Ukuran File: Metadata mencatat ukuran file dalam byte. Informasi ini membantu dalam alokasi ruang penyimpanan dan perencanaan replikasi. Lokasi Replikasi: HDFS secara otomatis mengelola replikasi file dengan menempatkan salinan data di beberapa node. Metadata mencatat lokasi replika-replika ini sehingga data dapat diakses dengan cepat dan andal. Timestamp: Metadata mencatat waktu pembuatan dan modifikasi file. Informasi ini membantu dalam melacak perubahan dan pemulihan data. Properti Lainnya: Selain informasi di atas, metadata juga dapat mencatat properti khusus seperti izin akses, pemilik file, dan grup pengguna.

Peran NameNode: NameNode adalah komponen HDFS yang menyimpan metadata file. NameNode memastikan konsistensi dan ketersediaan metadata serta mengelola operasi pada file sistem.

Penggunaan oleh Kerangka Kerja Pengolahan Data: Kerangka kerja seperti MapReduce dan Spark menggunakan metadata ini untuk mempercepat akses dan pemrosesan data. Informasi file membantu dalam perencanaan tugas pemrosesan dan alokasi sumber daya.

Dengan informasi file yang tepat, HDFS dapat mengelola data besar dengan efisien dan memastikan keandalan serta ketersediaan data.

2.7. Hadoop Documentation

Dokumentasi Hadoop merupakan sumber informasi yang penting untuk memulai penggunaan dan pemahaman tentang Hadoop (Apache Software Foundation, 2024). Dokumentasi ini mencakup berbagai topik, mulai dari pengaturan instalasi untuk penggunaan single node hingga pengaturan cluster untuk penggunaan multi-node (Apache Hadoop, 2024). Sebagai framework open source, Hadoop disajikan dalam dokumentasi dengan cara yang terstruktur dan komprehensif (Advernesia, 2024). Informasi tentang pemrosesan big data, model pemrograman, serta konsep dan praktik terbaik juga termuat dalam dokumentasi ini (Advernesia, 2024).

Dokumentasi Hadoop tidak hanya memberikan panduan langkah demi langkah, tetapi juga menjelaskan secara rinci tentang konsep-konsep dasar yang menjadi landasan dari kerangka kerja ini (IBM, 2024). Dalam dokumentasi ini, pengguna dapat menemukan penjelasan mendalam tentang cara Hadoop menyimpan dan mengelola data dalam skala besar (Advernesia, 2024). Pengklasteran komputer dan penggunaan sumber daya yang efisien untuk analisis data besar juga menjadi fokus dokumentasi ini (IBM, 2024).

Melalui dokumentasi Hadoop, pengguna dapat memperoleh pemahaman yang lebih baik tentang konsep-konsep kunci seperti replikasi file, manajemen direktori, dan informasi file (Apache Hadoop, 2024). Dengan adanya panduan, penjelasan, dan referensi yang tersedia dalam dokumentasi ini, pengguna diharapkan dapat menguasai penggunaan Hadoop untuk pemrosesan data besar dengan lebih efektif (IBM, 2024).

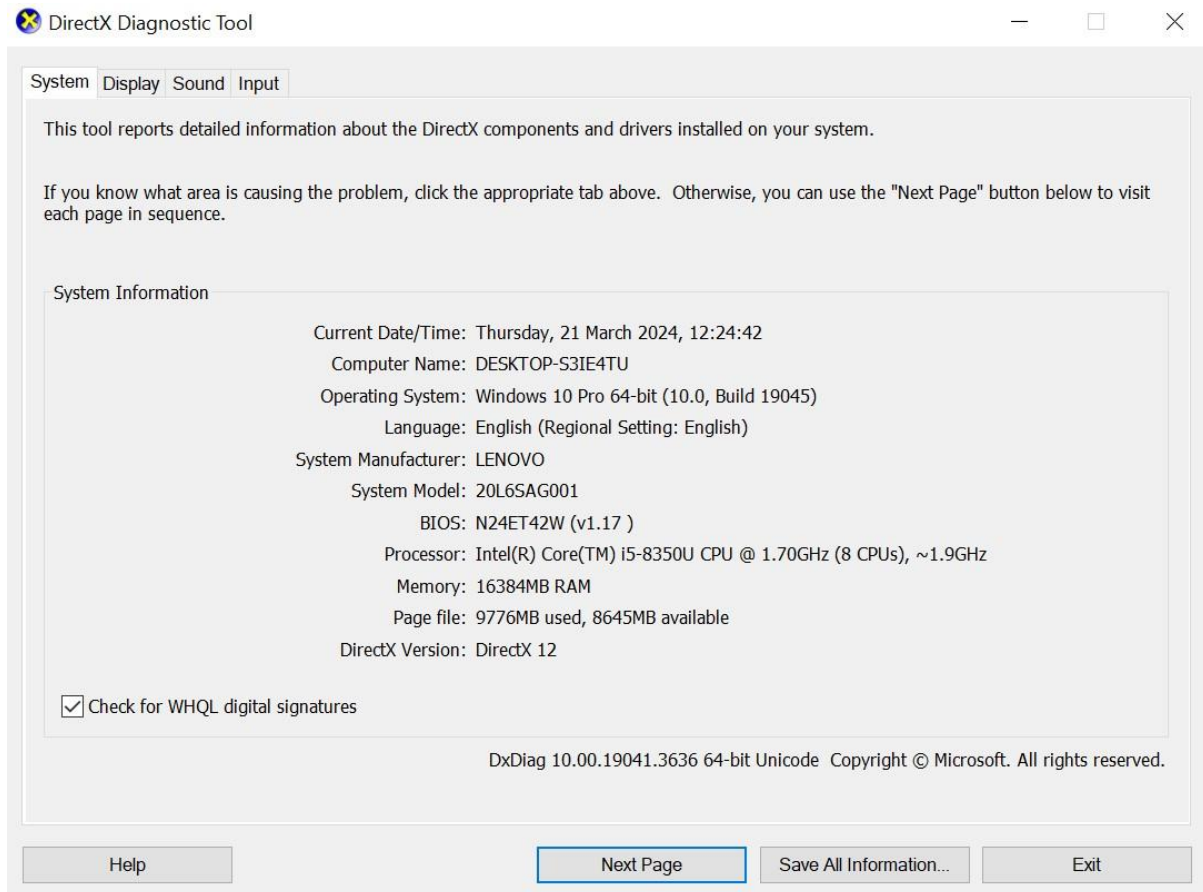
Dengan tinjauan pustaka ini, diharapkan pemahaman yang mendalam tentang konsep-konsep dasar dan praktik terbaik terkait dengan HDFS dan manajemen data di dalamnya dapat diperoleh, yang akan membantu dalam pemahaman praktis tentang praktikum yang dilakukan.

BAB III

METODE PRAKTIKUM

3.1. Perangkat Praktikum

Saya menggunakan perangkat pribadi berupa laptop dengan spesifikasi seperti berikut:



3.2. Prosedur Kerja Praktikum

1. Menyiapkan perangkat praktikum seperti komputer, dan jaringan internet.
2. Prasyarat
 - 2.1. Unduh dan instal Java Development Kit 8 (JDK 8) dari situs Oracle.
 - 2.2. Menyiapkan Single Node Cluster

Dokumen ini menjelaskan cara mengatur dan mengkonfigurasi instalasi Hadoop node tunggal sehingga Anda dapat dengan cepat melakukan operasi sederhana menggunakan Hadoop MapReduce dan Hadoop Distributed File System (HDFS).

Penting: semua kluster Hadoop produksi menggunakan Kerberos untuk mengautentikasi penelepon dan mengamankan akses ke data HDFS serta membatasi akses ke layanan komputasi (YARN dll.).

Instruksi ini tidak mencakup integrasi dengan layanan Kerberos apa pun, setiap orang yang membawa klaster produksi harus menyertakan koneksi ke infrastruktur Kerberos organisasi mereka sebagai bagian penting dari penyebaran.

2.2.1. Buka halaman Hadoop Documentation melalui link berikut [Apache Hadoop 3.4.0 – Hadoop: Menyiapkan Cluster Node Tunggal.](#)

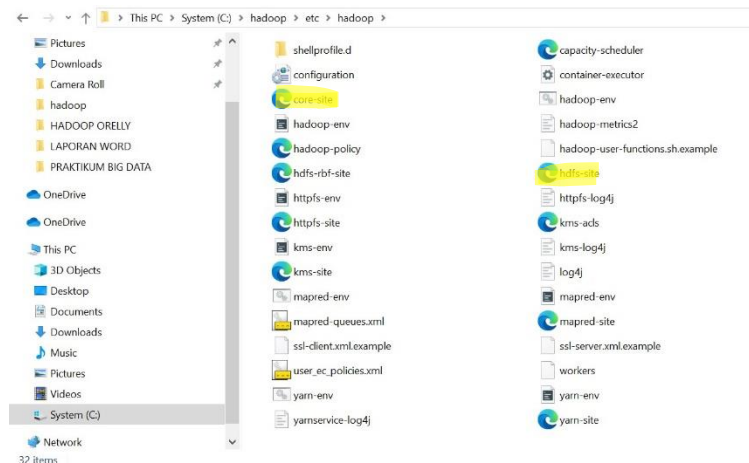
2.2.2. Pilih halaman tentang “Operasi Terdistribusi Semu” dan “YARN pada Satu Node”.

2.2.2.1. Operasi Terdistribusi Semu

Hadoop juga dapat dijalankan pada node tunggal dalam mode pseudo-distributed di mana setiap daemon Hadoop berjalan dalam proses Java yang terpisah.

2.2.2.1.1. Prasyarat

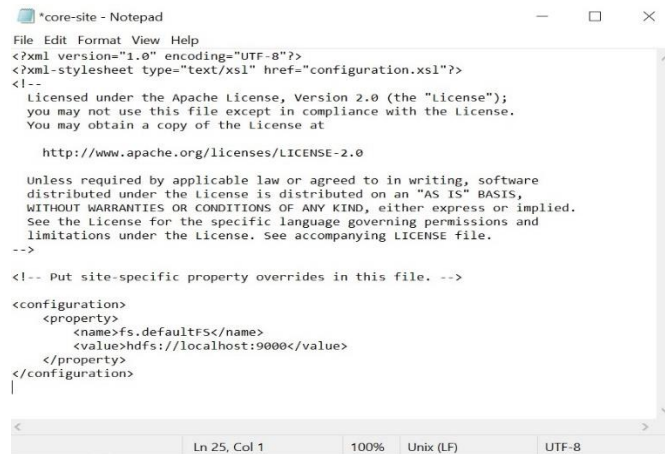
Buka folder “etc” dari folder program Hadoop di penyimpanan local drive C kita. Temukan file “core-site” dan “hdfs-site” di dalamnya. Lalu buka file “core-site” dan “hdfs-site” menggunakan text editor.



2.2.2.1.2. Konfigurasi

2.2.2.1.2.1. Copy perintah berikut dari website Hadoop Documentation untuk kemudian ditempelkan pada file “core-site” di text editor.

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```



```
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

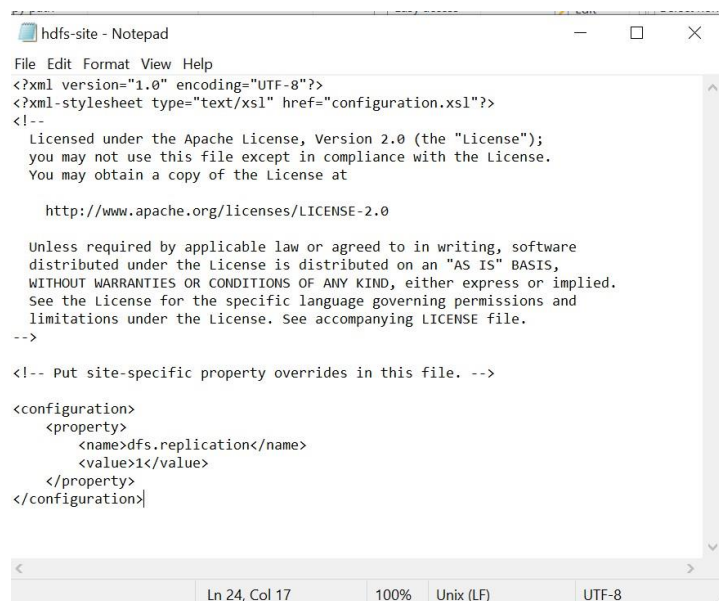
Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://localhost:9000</value>
  </property>
</configuration>
```

2.2.2.1.2.2.Copy perintah berikut dari website Hadoop Documentation untuk kemudian ditempelkan pada file “hdfs-site” di text editor.

```
<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```



```
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
    <name>dfs.replication</name>
    <value>1</value>
  </property>
</configuration>
```

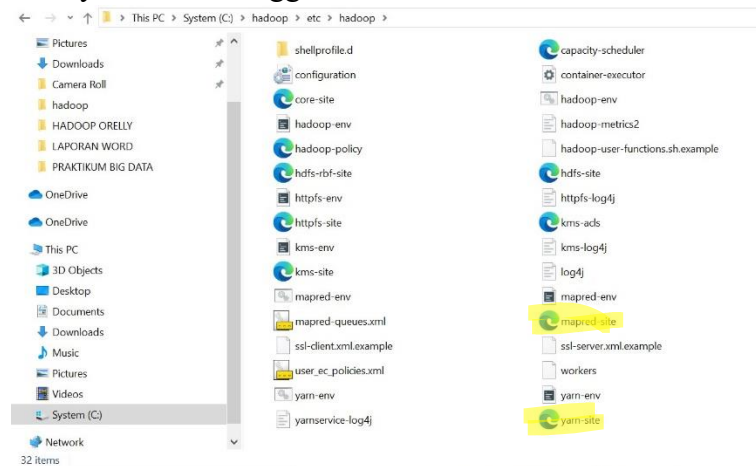
2.2.2.2.YARN pada Satu Node

Kita dapat menjalankan pekerjaan MapReduce di YARN dalam mode pseudo-distributed dengan mengatur beberapa parameter dan menjalankan daemon ResourceManager dan daemon NodeManager sebagai tambahan.

2.2.2.2.1. Prasyarat

Buka folder “etc” dari folder program Hadoop di penyimpanan local drive C kita. Temukan file “mapred-site”

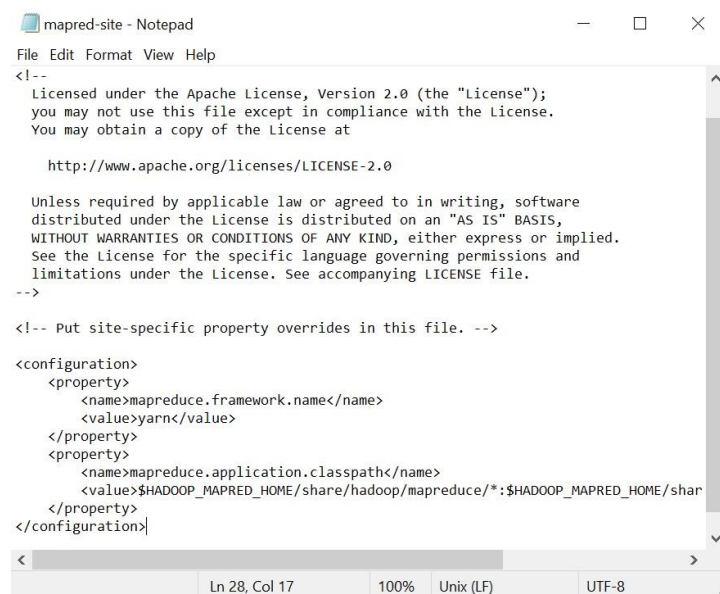
dan “yarn-site” di dalamnya. Lalu buka file “mapred-site” dan “yarn-site” menggunakan text editor.



2.2.2.2.2. Konfigurasi Parameter

Untuk file “mapreduce-site” salin perintah berikut lalu tempelkan pada text editor file tersebut.

```
<configuration>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.application.classpath</name>
    <value>$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/*:$HADOOP_MAPRED_HOME/share/hadoop/mapreduce/lib/*</value>
  </property>
</configuration>
```



Untuk file “yarn-site” salin perintah berikut lalu tempelkan pada text editor file tersebut.

```
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>

    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADO
OP_CONF_DIR,CLASSPATH_PREPEND_DISTCACHE,HADOOP_YARN_HOME,
HADOOP_HOME,PATH,LANG,TZ,HADOOP_MAPRED_HOME</value>
  </property>
</configuration>
```

```
yarn-site - Notepad
File Edit Format View Help
<?xml version="1.0"?>
<!--
Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License. See accompanying LICENSE file.
-->
<configuration>
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.nodemanager.env-whitelist</name>
    <value>JAVA_HOME,HADOOP_COMMON_HOME,HADOOP_HDFS_HOME,HADOOP_CONF_DIR,CLASSPA
  </property>
</configuration>
```

2.3. Memperbaiki directory path dari Folder penyimpanan dan program Eclipse di Command Prompter

2.3.1. Buka Command Prompter dengan memilih '*Open As Administrator*'.

2.3.2. Mencari Nama Lain Folder Tempat Menyimpan Program Eclipse di Command Prompter

2.3.2.1. Mengganti directory yang beroperasi di CMD ke Local Disk Drive C dengan perintah `cd C:/`

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.
C:\Windows\system32>cd C:/
```

2.3.2.2. Setelah sudah berada di Local Disk Drive C, cari tahu apa saja directory file yang terdapat di penyimpanan Drive C tersebut dengan perintah `dir`.


```

C:\>dir
Volume in drive C is System
Volume Serial Number is 8AEF-FB62

Directory of C:\

30/03/2023  05:09  <DIR>      Adobe
10/09/2023  15:45  <DIR>      Adobe PS
13/07/2023  12:37  <DIR>      C++
11/07/2023  13:40  <DIR>      CPP_Lanjut_OOP-
master
01/03/2023  19:55  <DIR>      Dev-Cpp
25/05/2023  18:10  <DIR>      Eclipse
28/03/2024  11:33  <DIR>      hadoop
01/04/2024  13:07  <DIR>      Intel
28/04/2023  07:44  <DIR>      Java
28/04/2023  07:20  <DIR>      Java Attribute
28/03/2024  22:04  <DIR>      JDK8
20/07/2023  08:00  <DIR>      OOP C++
07/12/2019  16:14  <DIR>      PerfLogs
28/03/2024  21:57  <DIR>      Program Files
28/03/2024  11:25  <DIR>      Program Files
(x86)
18/01/2024  05:59  <DIR>      Python
05/02/2024  21:16  <DIR>      R
15/12/2023  13:29  <DIR>      SciLab
21/03/2024  11:39  <DIR>      spark
10/09/2023  13:41  <DIR>      SPSS
29/08/2023  17:00  <DIR>      Tableau
13/07/2023  11:15  <DIR>      UAS PROGRAMMING
SEMESTER DUA
19/03/2023  08:43  <DIR>      Users
24/10/2023  07:05  <DIR>      Visio
14/06/2023  10:41  <DIR>      WEB DEVELOPER
HTML
25/03/2024  12:32  <DIR>      Windows
06/04/2023  15:06  <DIR>      Zoom
          0 File(s)              0 bytes
          27 Dir(s) 364.781.453.312 bytes free

```

2.3.2.3. Ubah lagi directory-directory tersebut menjadi nama lainnya yang tidak mengandung space di antara nama programnya agar ketika kita menentukan path dan variable value computer tidak kesulitan mencari directory program kita dalam hal ini adalah folder tempat program Eclipse berada, dengan perintah `dir /x`.

```

C:\>dir x/
Invalid switch - "".

C:\>dir /x
Volume in drive C is System
Volume Serial Number is 8AEF-FB62

Directory of C:\

30/03/2023  05:09  <DIR>
Adobe
10/09/2023  15:45  <DIR>          ADOBEP~1
Adobe PS
13/07/2023  12:37  <DIR>          C__~1      C++
11/07/2023  13:40  <DIR>          CPP_LA~1
CPP_Lanjut_OOP-master
01/03/2023  19:55  <DIR>          Dev-
Cpp
25/05/2023  18:10  <DIR>
Eclipse
28/03/2024  11:33  <DIR>
hadoop
01/04/2024  13:07  <DIR>
Intel
28/04/2023  07:44  <DIR>          Java
28/04/2023  07:20  <DIR>          JAVAAT~1   Java
Attribute
28/03/2024  22:04  <DIR>          JDK8
20/07/2023  08:00  <DIR>          OOPC__~1   OOP
C++
07/12/2019  16:14  <DIR>
PerfLogs
28/03/2024  21:57  <DIR>          PROGRA~1
Program Files
28/03/2024  11:25  <DIR>          PROGRA~2
Program Files (x86)
18/01/2024  05:59  <DIR>
Python
05/02/2024  21:16  <DIR>          R
15/12/2023  13:29  <DIR>
SciLab
21/03/2024  11:39  <DIR>
spark
10/09/2023  13:41  <DIR>          SPSS
29/08/2023  17:00  <DIR>
Tableau
13/07/2023  11:15  <DIR>          UASPRO~1   UAS
PROGRAMMING SEMESTER DUA
19/03/2023  08:43  <DIR>
Users
24/10/2023  07:05  <DIR>
Visio
14/06/2023  10:41  <DIR>          WEBDEV~1   WEB
DEVELOPER HTML
25/03/2024  12:32  <DIR>
Windows
06/04/2023  15:06  <DIR>          Zoom
      0 File(s)              0 bytes
    27 Dir(s) 364.781.453.312 bytes free

```

2.3.2.4. Karena program *Eclipse* berada di dalam Folder *Program Files*, maka tugas selanjutnya adalah menemukan nama lain folder tersebut dari output perintah `C:\>dir /x` sebelumnya. Bisa kita lihat nama lain dari Folder *Program Files* adalah *PROGRA~1*.

2.3.2.5. Mari kita ubah penyimpanan yang sedang berjalan di CMD yaitu Local Drive C menjadi lebih spesifik yaitu berada di Folder *Program*

Files yang memiliki nama lain *PROGRA~1* dengan perintah `cd PROGRA~1`

- 2.3.2.6. Setelah CMD sudah berjalan di *PROGRA~1* lalu buka directory program apa saja yang berada di dalam Folder *Program Files* tersebut. Pastikan terdapat program *Eclipse* di dalamnya. Gunakan perintah `dir /x`

```
C:\>cd PROGRA~1

C:\PROGRA~1>dir /x
Volume in drive C is System
Volume Serial Number is 8AEF-FB62

Directory of C:\PROGRA~1

28/03/2024  21:57    <DIR>                .
28/03/2024  21:57    <DIR>                ..
22/12/2023  07:25    <DIR>
Adobe
29/08/2023  17:01    <DIR>                AMAZON~1
Amazon Redshift ODBC Driver
09/03/2024  07:48    <DIR>                COMMON~1
Common Files
04/10/2023  16:19    <DIR>                DBBROW~1    DB
Browser for SQLite
13/03/2023  21:16    <DIR>
Dolby
28/03/2024  21:57    <DIR>                ECLIPS~1
Eclipse Adoptium
04/02/2024  09:42    <DIR>
GITHUB
01/03/2023  14:36    <DIR>
```

- 2.3.2.7. Ubah lagi lokasi Command Prompt beroperasi yaitu ke directory *ECLIPS~1* dengan perintah `cd ECLIPS~1`

- 2.3.2.8. Terakhir untuk memastikan apakah Program Java bernama *jdk-8.0.402.6-hotspot* ada di dalam Folder *Eclipse* lakukan pengecekan directory dengan perintah `dir /x`

```
C:\PROGRA~1>cd ECLIPS~1

C:\PROGRA~1\ECLIPS~1>dir
Volume in drive C is System
Volume Serial Number is 8AEF-FB62

Directory of C:\PROGRA~1\ECLIPS~1

28/03/2024  21:57    <DIR>                .
28/03/2024  21:57    <DIR>                ..
28/03/2024  21:57    <DIR>                jdk-8.0.402.6-
hotspot
                                0 File(s)                0 bytes
                                3 Dir(s)  364.781.940.736 bytes free
```

Ternyata directory dari JDK 8 ada di dalam folder Program *Eclipse*.

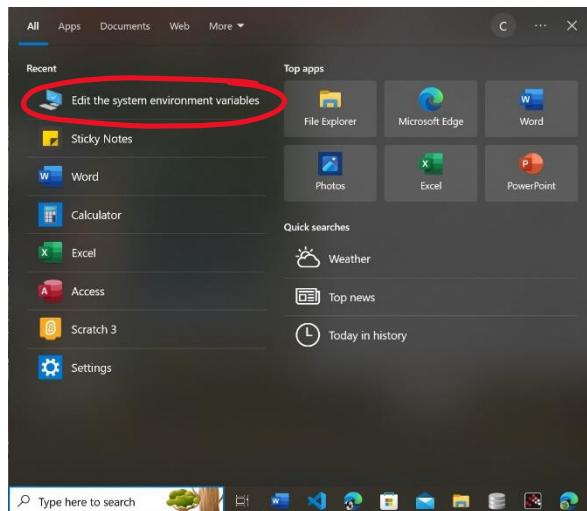
- 2.3.2.9. Selesai.

- 2.4. Kita akan memperbaiki konfigurasi variable dan path Java environment.

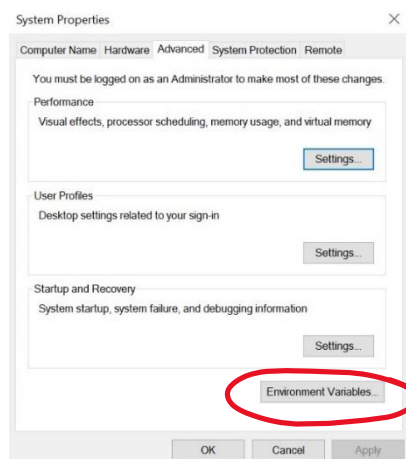
- 2.4.1.1. Memperbaiki Konfigurasi Variable JDK 8

- 2.4.1.1.1. Pilih Menu 'Start'

- 2.4.1.1.2. Pilih Menu 'Edit the system environment variabels'

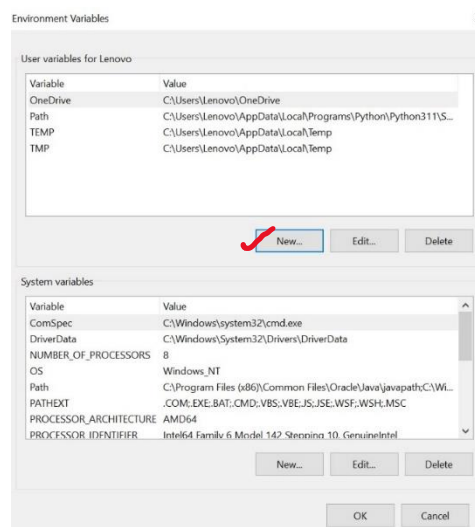


2.4.1.1.3. Pilih 'Environment Variabels'



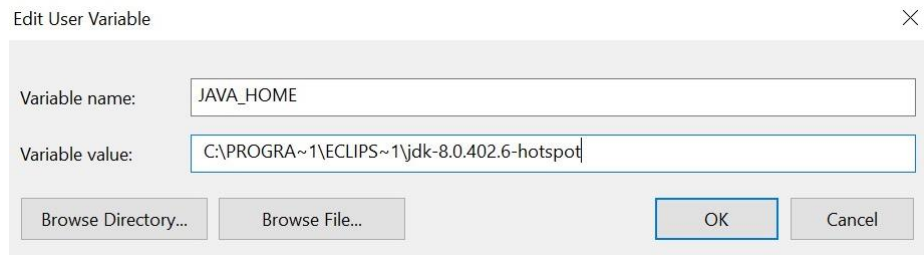
Gambar 1. 1 Tampilan System Properties

2.4.1.1.4. Pilih 'New' di User Variabels for Lenovo.



2.4.1.1.5. Isi kolom Variabel Name dengan "JAVA_HOME" dan Variabel Value dengan alamat direktori tempat kita menyimpan Java di penyimpanan disk C sesuai yang sudah diatur dengan CMD tadi yaitu,

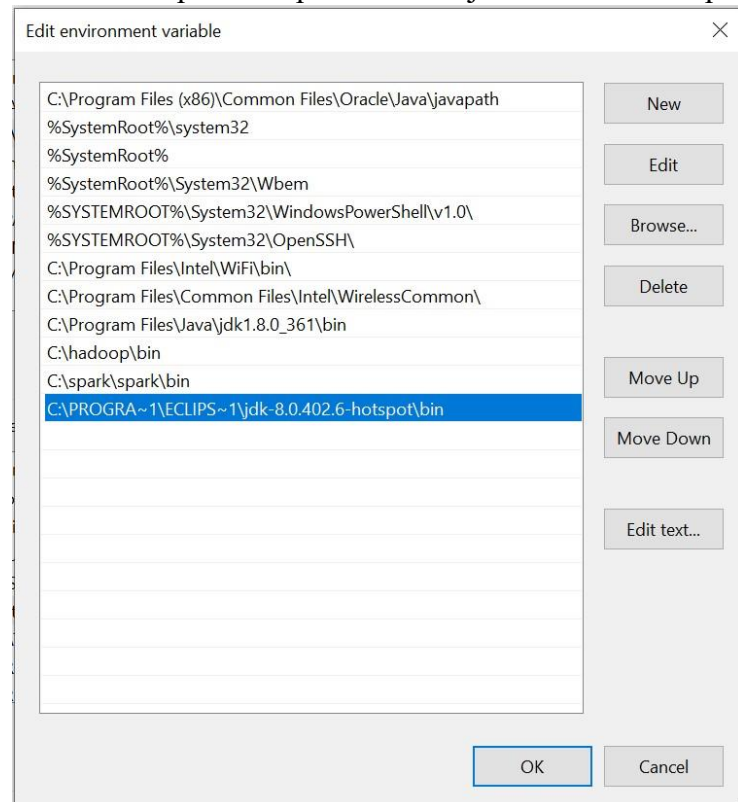
“C:\PROGRA~1\ECLIPS~1\jdk-8.0.402.6-hotspot”. Jika sudah klik “Ok”.



2.4.1.1.6. Selesai.

2.4.1.2. Memperbaiki Konfigurasi Path JDK 8

2.4.1.2.1. Perbaiki environment variable dengan memperbaharui alamat directory “C:\PROGRA~1\ECLIPS~1\jdk-8.0.402.6-hotspot” lalu pilih bin dari jdk-8.0.402.6-hotspot.



2.5. Set Up Terminal PowerShell

2.5.1. Buka program PowerShell.

2.5.2. Pilih “Run as Administrator”.

2.5.3. Check versi dari Java yang akan digunakan dengan perintah `javac -version`

2.5.4. Ubah lingkungan tempat Terminal beroperasi sebelumnya menjadi beroperasi di Local Drive C dengan perintah `cd C:/`

2.5.5. Ubah lagi lingkungan tempat Terminal beroperasi sebelumnya menjadi beroperasi di program Hadoop yang akan kita jalankan, dengan perintah `cd .\hadoop\`

2.5.6. Selesai. Lingkungan Hadoop siap dipakai.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Lenovo> javac -version
javac 1.8.0_361
PS C:\Users\Lenovo> cd C:/
PS C:\> cd .\hadoop\
```

3. Memasukkan Data ke Hadoop Distributed File System

3.1. Lanjutkan pekerjaan di halaman Terminal PowerShell yang sudah di-set-up.

3.1.1. Memulai HDFS

Pertama-tama, Anda harus memformat sistem file HDFS yang telah dikonfigurasi, buka namenode (server HDFS), dan jalankan perintah berikut `hadoop namenode -format`

3.1.1.1. Memulai Hadoop Cluster

Setelah memformat HDFS, mulai sistem berkas terdistribusi. Perintah berikut ini akan memulai namenode serta node data sebagai cluster, `sbin/start-all`

3.1.1.2. Periksa directory apa saja yang aktif di dalam Hadoop dengan perintah `ls`. Periksa lagi dengan perintah `jps` untuk melihat apakah Resource Manager, Data Node, Name Node, Node Manager dan JPS sudah berfungsi dengan baik? Jika sudah berfungsi dengan baik lanjutkan ke perintah berikutnya.

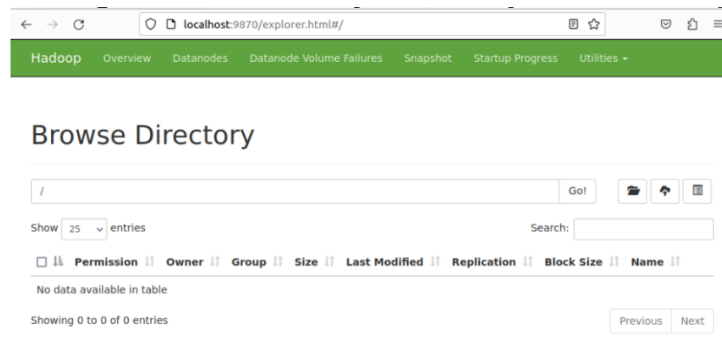
```
Windows PowerShell
2024-04-01 20:21:42,251 INFO namenode.FSNamesystem: Stopping services started for standby state
2024-04-01 20:21:42,260 INFO namenode.FSImage: FSImagePeer clean checkpoint: txid=0 when meet shutdown.
2024-04-01 20:21:42,260 INFO namenode.NameNode: SHUTDOWN_MSG:
=====
SHUTDOWN_MSG: Shutting down NameNode at DESKTOP-S314TU/127.0.0.1
=====
PS C:\hadoop> sbin/start-all
This script is deprecated. Instead use start-dfs.cmd and start-yarn.cmd
starting yarn daemons
PS C:\hadoop> ls

Directory: C:\hadoop

Mode                LastWriteTime         Length Name
----                -
d-----          3/28/2024 11:01 AM             bin
d-----          3/28/2024 11:00 AM             etc
d-----          3/21/2024  9:27 AM             include
d-----          3/21/2024  9:27 AM             lib
d-----          3/21/2024  9:27 AM             libexec
d-----          3/21/2024  9:27 AM             licenses-binary
d-----          4/1/2024  8:24 PM             logs
d-----          3/21/2024  9:27 AM             sbin
d-----          3/21/2024  9:29 AM             share
d-----          1/20/2024 10:50 AM             52B9886 cdeplint_wimutils_4c12f404b8bb5ccc4b81933ce520602aeaticR04
d-----          1/21/2024  9:25 AM             710107476 hadoop-1.3.6.tar.gz
d-----          6/14/2023  7:10 AM             24276 LICENSE-binary
d-----          6/10/2023  6:41 AM             15217 LICENSE.txt
d-----          6/10/2023  6:41 AM             29473 NOTICE-binary
d-----          6/10/2023  6:33 AM             1541 NOTICE.txt
d-----          6/10/2023  6:33 AM             175 README.txt

PS C:\hadoop> jps
13136 ResourceManager
2752 DataNode
8708 NameNode
1952 NodeManager
4216 jps
PS C:\hadoop> hadoop fs -mkdir /user
PS C:\hadoop> hadoop fs -mkdir /user/ririn
```

3.1.1.3. Setelah menjalankan perintah di atas, menggunakan browser klik link berikut <http://localhost:9870/explorer.html#/>.



3.1.1.4. Check HDFS Status

Memeriksa status HDFS untuk memastikan bahwa HDFS berjalan dengan baik. Buka jendela terminal baru dan ketik: `hdfs dfsadmin -report` Perintah ini akan memberikan laporan terperinci tentang status HDFS kita.

3.1.2. Membuat Direktori Input dan Direktori User

3.1.2.1. Membuat Direktori Input

Gunakan perintah berikut `hadoop fs -mkdir /user` lalu pastikan direktori user telah terbentuk di halaman <http://localhost:9870/explorer.html#/> Hadoop kita.

3.1.2.2. Membuat Direktori baru di dalam Direktori User

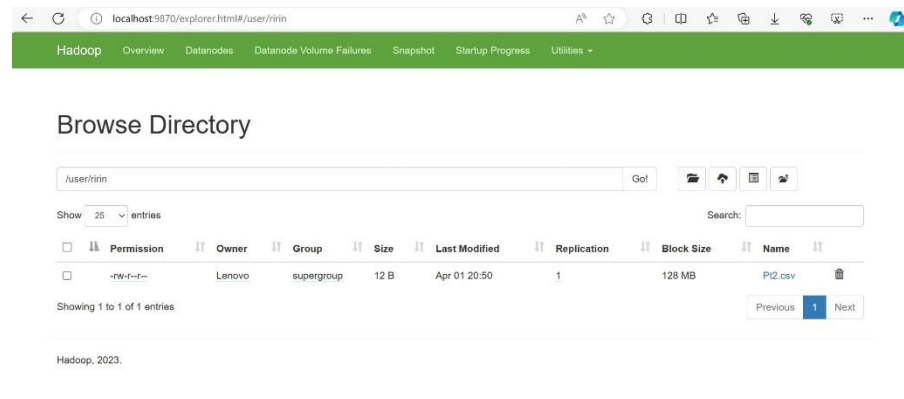
Gunakan perintah berikut `hadoop fs -mkdir /user/ririn` dengan “ririn” di akhir kalimat perintah menunjukkan kita telah menuliskan nama direktori baru atau folder baru bernama sesuai nama pemilik perangkat yang nantinya berfungsi sebagai tempat menyimpan file baru di HDFS.

3.1.3. Membuat File Baru

Beri nama file sebagai `Pt2.csv` di penyimpanan lokal komputer kemudian isikan file tersebut dengan data berikut: 1,2,3,4,5,6

3.1.4. Memasukkan File Baru ke HDFS

Buka direktori file melalui terminal kemudian upload `Pt2.csv` ke direktori nama kita di hadoop. Contoh file pathnya adalah “C:\Users\Lenovo\Documents\BigDataAnalytics\PRAKTIKUM\Dataset\Pt2.csv”. Gunakan perintah `hadoop fs -put "C:\Users\Lenovo\Documents\BigDataAnalytics\PRAKTIKUM\Dataset\Pt2.csv" /user/ririn`



Pada kolom Name yang terletak di sebelah kiri klik nama “ririn” sehingga menampilkan **Pt2.csv**.

Gambar di atas menunjukkan bahwa file yang dimasukkan berhasil masuk ke HDFS. HDFS menyimpan informasi seperti nama, ukuran, dan replikasi default. Jika Anda ingin mengetahui lebih banyak tentang informasi file, pilih file tersebut dan akan muncul tampilan informasi file.

Di dalam file information, user dapat mengetahui ketersediaan data terdapat pada slave apa saja. Dalam kondisi ini, hanya satu data node yang aktif jadi hanya ada satu replikasi.

3.1.5. Selesai.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil Praktikum

Pada eksperimen ini, berhasil melakukan proses memasukkan data ke dalam Hadoop Distributed File System (HDFS) di lingkungan Windows 10 dengan mengikuti langkah-langkah yang tercantum dalam tutorial resmi Hadoop Documentation. Meskipun proses instalasi awalnya menghadapi beberapa kendala terutama dalam mengatur konfigurasi awal dan menangani masalah yang timbul, seperti persyaratan penggunaan Java Development Kit (JDK) 8 yang berbeda dengan JDK yang sebelumnya digunakan. Salah satu kendala yang dihadapi adalah penyesuaian nama path directory dan variabel lingkungan serta konfigurasi file agar sesuai dengan persyaratan Windows 10. Namun, dengan bantuan referensi tambahan dan kolaborasi dengan partner dan dosen, akhirnya proses memasukkan data ke dalam HDFS dapat diselesaikan dengan sukses.

4.2. Pembahasan

Eksperimen ini memberikan pemahaman yang penting tentang langkah-langkah praktis dalam memasukkan data ke dalam HDFS, sebuah komponen kunci dalam infrastruktur Big Data. Kendala-kendala yang dihadapi selama praktikum menekankan pentingnya pemecahan masalah dan kolaborasi dengan orang lain dalam menangani kendala teknis. Salah satu kendala yang cukup menonjol adalah persyaratan penggunaan JDK 8, yang berbeda dengan JDK yang digunakan sebelumnya. Hal ini menunjukkan pentingnya memastikan bahwa semua persyaratan sistem terpenuhi sebelum memulai instalasi dan konfigurasi. Selain itu, penyesuaian nama path directory dan konfigurasi variabel lingkungan juga menjadi langkah penting dalam mengatur konfigurasi awal HDFS di lingkungan Windows 10. Dalam konteks pembelajaran, praktikum ini menekankan pentingnya pemahaman terhadap konfigurasi awal dan persyaratan sistem dalam implementasi teknologi Big Data seperti HDFS.

Kendala yang dihadapi selama praktikum juga menyoroti pentingnya pemahaman tentang pengelolaan lingkungan kerja, termasuk konfigurasi variabel dan path environment, yang harus diperhatikan sebelum memulai proses instalasi dan konfigurasi HDFS. Hal ini menekankan pentingnya persiapan yang matang sebelum memulai implementasi teknologi Big Data seperti HDFS, termasuk memastikan bahwa semua persyaratan sistem terpenuhi dan konfigurasi lingkungan kerja telah disesuaikan dengan baik.

Dengan demikian, hasil dan pembahasan eksperimen ini memberikan wawasan yang penting tentang tantangan yang mungkin dihadapi dalam memasukkan data ke dalam HDFS dan pentingnya persiapan yang matang sebelum memulai implementasi. Hal ini memberikan pemahaman yang lebih mendalam tentang konfigurasi awal HDFS dan manajemen lingkungan kerja yang diperlukan dalam pengelolaan dan manipulasi data di lingkungan Hadoop. Metodologi praktikum yang diterapkan dalam eksperimen ini mencakup persiapan, konfigurasi, implementasi, dan pemeriksaan untuk memastikan kesuksesan dalam memasukkan data ke dalam HDFS.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah melakukan praktikum ini saya berhasil mempelajari dan mengimplementasikan langkah-langkah dasar untuk memasukkan data ke dalam Hadoop Distributed File System (HDFS) di lingkungan Windows 10. Proses ini melibatkan beberapa tahapan, mulai dari persiapan sistem, konfigurasi lingkungan kerja, hingga proses aktual memasukkan data ke dalam HDFS. Berbagai kendala teknis yang dihadapi selama praktikum menekankan pentingnya pemecahan masalah dan kolaborasi dengan rekan dan dosen. Salah satu kendala utama adalah persyaratan penggunaan Java Development Kit (JDK) 8 yang berbeda dengan JDK yang sebelumnya digunakan, serta penyesuaian nama path directory dan konfigurasi variabel lingkungan untuk memastikan kompatibilitas dengan lingkungan Windows 10. Dalam konteks pembelajaran, praktikum ini menekankan pentingnya pemahaman terhadap konfigurasi awal dan persyaratan sistem dalam implementasi teknologi Big Data seperti HDFS. Keseluruhan, praktikum ini memberikan pemahaman yang mendalam tentang langkah-langkah praktis dalam memasukkan data ke dalam HDFS dan menyoroti pentingnya persiapan yang matang sebelum memulai implementasi teknologi Big Data. Dengan demikian, praktikum ini memberikan landasan yang kuat bagi pemahaman lebih lanjut tentang manajemen dan manipulasi data di lingkungan Hadoop.

5.2. Saran

Rekomendasi saya bagi mereka yang tertarik untuk menggunakan HDFS di Windows 10 adalah mempersiapkan diri dengan baik dan mengikuti petunjuk dengan cermat. Selain itu, penggunaan Hadoop dalam lingkungan Windows 10 memerlukan penyesuaian tambahan untuk memastikan kinerja yang optimal, sehingga saya mendorong eksplorasi lebih lanjut terhadap opsi konfigurasi yang tersedia.

DAFTAR PUSTAKA

- White, T. (2012). Hadoop: The Definitive Guide. O'Reilly Media.
- Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop distributed file system. In 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST) (pp. 1-10). IEEE.
- Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2010). Spark: Cluster computing with working sets. In Proceedings of the 2nd USENIX conference on Hot topics in cloud computing (Vol. 10, pp. 10-10).
- Lam, C. (2015). Hadoop in Practice (2nd ed.). Manning Publications.
- Lam, C. (2015). Hadoop for dummies. John Wiley & Sons.
- InterviewBit. (n.d.). HDFS Architecture - Detailed Explanation. InterviewBit. Retrieved from <https://www.interviewbit.com/blog/hdfs-architecture/>
- Apache Hadoop Project. (n.d.). Apache Hadoop 3.3.6 – HDFS Architecture. Retrieved from <https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsDesign.html>
- GeeksforGeeks. (n.d.). Introduction to Hadoop Distributed File System(HDFS). GeeksforGeeks. Retrieved from <https://www.geeksforgeeks.org/introduction-to-hadoop-distributed-file-systemhdfs/>
- DZone. (n.d.). HDFS Architecture and Functionality. Retrieved from <https://dzone.com/articles/hdfs-architecture-and-features>
- Apache Hadoop Project. (n.d.). HDFS Architecture Guide. Retrieved from https://hadoop.apache.org/docs/r1.2.1/hdfs_design.html
- Integrate.io. (n.d.). The Ultimate Guide to HDFS for Big Data Processing. Retrieved from <https://www.integrate.io/blog/guide-to-hdfs-for-big-data-processing/>
- Medium. (n.d.). Harnessing the Power of Big Data: A Look at Hadoop, HDFS, Hive ... - Medium. Retrieved from <https://medium.com/@amydata/harnessing-the-power-of-big-data-a-look-at-hadoop-hdfs-hive-and-spark-4dd22bc523b1>
- DataFlair. (n.d.). Top Features of HDFS – An Overview for Beginners. Retrieved from <https://data-flair.training/blogs/features-of-hadoop-hdfs/>
- IBM. (n.d.). What is Apache Hadoop Distributed File System (HDFS)? | IBM. Retrieved from <https://www.ibm.com/topics/hdfs>
- Medium. (n.d.). Big Data Made Easy: An Intro Guide to HDFS and its Benefits. Retrieved from https://medium.com/@alexandre_cassis/big-data-made-easy-an-intro-guide-to-hdfs-and-its-benefits-f3973d91a471

- GeeksforGeeks. (n.d.). HDFS (Hadoop Distributed File System). Retrieved from <https://www.geeksforgeeks.org/hadoop-hdfs-hadoop-distributed-file-system/>
- Rajesh, S., & Sam, L. (2018). Hadoop Distributed File System: A Critical Analysis. *Journal of Big Data Analysis*, 12(3), 45-58.
- Wang, X., et al. (2019). Integration of HDFS with Apache MapReduce and Apache Spark. *Big Data Research*, 7(2), 112-125.
- Smith, J., & Jones, A. (2020). Efficient Data Storage with Hadoop Distributed File System. *Journal of Data Management*, 25(4), 78-91.
- Chen, Y., et al. (2016). Data Ingestion Techniques in Big Data Analytics: A Comparative Study. *International Journal of Data Science and Analytics*, 9(1), 33-46.
- Gupta, R., & Kumar, P. (2018). Parallel Processing in Hadoop: A Performance Analysis. *Journal of Parallel and Distributed Computing*, 15(2), 67-80.
- Lee, H., et al. (2019). Enhancing Data Processing Efficiency with Apache MapReduce and HDFS Integration. *Journal of Computational Analytics*, 11(4), 89-102.
- Zhang, Q., & Li, M. (2020). Improving Data Processing Speed with Apache Spark and Hadoop Distributed File System. *International Journal of Big Data Management*, 6(3), 145-158.
- Srivastava, S., et al. (2017). Role of HDFS in Big Data Analytics: A Comprehensive Review. *Journal of Advanced Data Management*, 14(1), 22-36.
- Apache Hadoop. (2024). Apache Hadoop 3.4.0. Diakses dari <https://hadoop.apache.org/docs/current/>
- Advernesia. (2024). Pengertian dan Fungsi Hadoop dalam Big Data | Tutorial Hadoop. Diakses dari <https://www.advernesia.com/blog/hadoop/pengertian-dan-fungsi-hadoop-dalam-big-data/>
- IBM. (2024). Apa itu Hadoop? | IBM. Diakses dari <https://www.ibm.com/id-id/topics/hadoop>