

# LAPORAN PRAKTIKUM

## PRAKTIKUM BIG DATA ANALYTICS “INSTALASI APACHE HADOOP 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-1 mata kuliah Big Data Analytics. Praktikum ini menghadirkan kesempatan bagi saya untuk menginstalasi dan mengonfigurasi Apache Hadoop di lingkungan Windows 10, sebuah langkah teknis yang esensial dalam memahami fondasi analisis data besar.

Laporan ini mencerminkan hasil dari upaya saya dalam mengeksplorasi proses instalasi dan konfigurasi 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
DAFTAR GAMBAR.....	iii
BAB I .....	1
PENDAHULUAN .....	1
1.1. Latar Belakang Praktikum.....	1
1.2. Tujuan Praktikum.....	1
BAB II.....	2
TINJAUAN PUSTAKA.....	2
2.1. Konsep Big Data.....	2
2.2. Apache Hadoop.....	7
2.3. Instalasi dan konfigurasi Hadoop.....	13
2.4. Apache Spark.....	14
2.5. Kompatibilitas Apache Hadoop dan Apache Spark dengan Windows 10.....	17
BAB III.....	19
METODE PRAKTIKUM .....	19
3.1. Perangkat Praktikum.....	19
3.2. Prosedur Kerja Praktikum.....	19
BAB IV.....	33
HASIL DAN PEMBAHASAN.....	33
4.1. Hasil Praktikum .....	33
4.2. Pembahasan .....	33
BAB V .....	35
PENUTUP .....	35
5.1. Kesimpulan.....	35
5.2. Saran.....	35
DAFTAR PUSTAKA.....	36

# DAFTAR GAMBAR

Gambar 1. 1 Definisi Big Data.....	2
Gambar 1. 2 Karakteristik Big Data.....	3
Gambar 1. 3 Contoh Pemanfaatan Big Data Untuk Penawaran Mcdonald's .....	5
Gambar 1. 4 Contoh Pemanfaatan Big Data di Aplikasi PeduliLindungi .....	6
Gambar 1. 5 Perbedaan Apache Hadoop vs Apache Spark .....	17
Gambar 1. 6 Spesifikasi Perangkat Komputer yang digunakan untuk Praktikum .....	19
Gambar 1. 7 Menu Start Windows.....	20
Gambar 1. 8 Tampilan System Properties .....	20
Gambar 1. 9 Tampilan Environment Variables.....	20
Gambar 1. 10 Tampilan New User Variables untuk Java .....	21
Gambar 1. 11 Tampilan Environtment Variables untuk JAVA_HOME .....	21
Gambar 1. 12 Tampilan Website Apache Hadoop .....	22
Gambar 1. 13 Proses Downloading Apache Hadoop.....	22
Gambar 1. 14 Proses Extracting Apache Hadoop.....	22
Gambar 1. 15 Ubah Nama Folder Extract Hadoop .....	23
Gambar 1. 16 Tampilan Website Apache Spark .....	23
Gambar 1. 17 Tampilan Website Channel untuk Mengunduh Apache Spark .....	24
Gambar 1. 18 . Downloading Apache Spark .....	24
Gambar 1. 19 Peringatan Pemindahan Saat Akan Mengekstrak Folder Apache Spark di Penyimpanan Lokal Disk C.....	24
Gambar 1. 20 Extracting Folder Apache Spark.....	25
Gambar 1. 21 Mengganti Nama Folder Ekstraksi Apache Spark.....	25
Gambar 1. 22 Menyunting User Variabel dengan JAVA_HOME .....	25
Gambar 1. 23 Menyunting User Variabel dengan HADOOP_HOME.....	26
Gambar 1. 24 Menyunting User Variabel dengan SPARK_HOME.....	26
Gambar 1. 25 Daftar User Variabel untuk Java, Hadoop dan Spark .....	27
Gambar 1. 26 Seleksi Menu Path.....	27
Gambar 1. 27 Menyunting Environment Variabel untuk Path Java.....	28
Gambar 1. 28 Seleksi bin Java .....	28
Gambar 1. 29 Seleksi bin Spark.....	29
Gambar 1. 30 Seleksi bin Hadoop.....	29
Gambar 1. 31 Hasil Membuat Path Java, Hadoop dan Spark .....	30
Gambar 1. 32 Menjalankan Hadoop di CMD .....	30
Gambar 1. 33 Menjalankan Spark di CMD.....	31
Gambar 1. 34 Membuka Spark Melalui Surel Local Host pada CMD .....	31
Gambar 1. 35 Tampilan Spark di Website .....	31
Gambar 1. 36 Tampilan Menu Executors Spark di Website .....	32

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang Praktikum

Pemanfaatan teknologi Big Data sedang mengalami pertumbuhan yang cepat di berbagai bidang, mulai dari sektor bisnis dan keuangan hingga dalam bidang ilmu pengetahuan dan kesehatan dengan produksi data yang diprediksi 44 kali lebih besar dari tahun 2009 (Khan et al., 2014). Karena meningkatnya jumlah data yang terus mengalir, big data menghadapi tantangan terkait load balancing yang tepat, penyimpanan dan pemrosesan file kecil, dan deduplikasi (Dev & Patgiri, 2016) maka diperlukanlah suatu kerangka kerja yang mampu mengelola dan menganalisis data dalam skala besar. Di sinilah peran utama Apache Hadoop menjadi sangat penting. Apache Hadoop adalah platform pilihan untuk mengembangkan aplikasi intensif data skala besar, yang memungkinkan pengguna untuk mengolah beberapa terabyte data menggunakan empat hingga 4.000 komputer (Bhandarkar, 2010). Sebagai sebuah platform open-source yang secara khusus dikembangkan untuk mengelola dan menganalisis dataset besar secara efisien, memungkinkan analisis statistik lengkap dan toleransi kesalahan dalam aplikasi (Mall & Rana, 2016), Apache Hadoop telah menjadi fondasi bagi banyak solusi terkait Big Data pada zaman sekarang.

Meskipun popularitas Apache Hadoop telah meluas, implementasinya sering kali dihadapkan pada berbagai tantangan teknis, terutama dalam hal instalasi dan konfigurasi. Tantangan ini muncul karena kompleksitas dalam menyiapkan infrastruktur yang dibutuhkan dan masalah kompatibilitas dengan sistem operasi tertentu. Terutama, dengan dominasi Linux sebagai platform yang paling sering digunakan dalam lingkungan Big Data, pemasangan Apache Hadoop di lingkungan Windows 10 menjadi langkah yang menarik untuk dieksplorasi lebih lanjut.

Karena itulah, tujuan dari praktikum ini adalah untuk memberikan pemahaman yang praktis tentang proses instalasi dan konfigurasi Apache Hadoop pada lingkungan Windows 10. Selain itu, praktikum ini juga bertujuan untuk memperluas pemahaman teknis mahasiswa dalam mengelola infrastruktur Big Data serta untuk mengeksplorasi cara implementasi ini dapat diterapkan dalam pengembangan aplikasi dan analisis data di dunia nyata.

### 1.2. Tujuan Praktikum

Tujuan dari praktikum ini adalah sebagai berikut:

1. Mahasiswa dapat memahami proses instalasi Apache Hadoop di lingkungan Windows 10.
2. Mahasiswa dapat mengkonfigurasi Apache Hadoop agar berjalan dengan baik di platform Windows 10.
3. Mahasiswa dapat memperluas pemahaman mereka tentang infrastruktur Big Data dan penerapannya dalam analisis data.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Konsep Big Data

##### 2.1.1. Definisi Big Data

### Big Data

Kumpulan data yang jumlahnya besar dan kompleks. Big data bisa meliputi data terstruktur, semi-terstruktur, dan tidak terstruktur.



<https://revou.co/revoupedia/kosakata>  
Sumber: Oracle Cloud Infrastructure

*Gambar 1. 1 Definisi Big Data*

Menurut Oracle Cloud Infrastructure, big data adalah himpunan data yang memiliki volume besar dan kompleksitas yang tinggi. Data ini terus berkembang dengan cepat dari waktu ke waktu.

Big data dapat mencakup berbagai jenis data, termasuk data terstruktur, semi-terstruktur, dan tidak terstruktur. Data terstruktur sering kali memiliki format yang terorganisir dengan baik, sementara data semi-terstruktur mencakup unsur dari kedua jenis struktur data tersebut. Data tidak terstruktur dapat memiliki format yang beragam dan sulit diukur.

Biasanya, big data disimpan dalam basis data komputer dan dianalisis menggunakan perangkat lunak khusus. Hasil analisis big data sering digunakan untuk membantu dalam pengambilan keputusan bisnis.

### 2.1.2. Karakteristik Big Data



Gambar 1. 2 Karakteristik Big Data

#### a. Volume

Nama "big data" mengartikan data yang memiliki kuantitas atau ukuran sangat besar. Ukuran itu disebut sebagai volume. Artinya, penentuan apakah kumpulan data tertentu bisa disebut big data atau tidak tergantung pada berapa volume-nya.

#### b. Velocity

Salah satu aspek utama big data adalah menyediakan data secara real-time dalam waktu cepat. Velocity berkaitan dengan kecepatan perusahaan dalam menerima, menyimpan, dan mengelola data tersebut.

#### c. Variety

Variety mengacu pada sumber data yang heterogen (beragam). Saat dikumpulkan dari berbagai sumber, data memiliki format, jenis, dan kategori berbeda, meliputi:

- Structured data – data yang memiliki skema terorganisasi dan disimpan dalam relational database management system.
- Semi-structured data – Perpaduan antara structured dan unstructured data. Jenis data ini tidak ada dalam relational database tetapi memiliki beberapa properti yang lebih mudah dianalisis dibanding unstructured data.
- Unstructured data – data yang tidak bisa disimpan di baris dan kolom relational database, seperti teks, audio, gambar, dan video.

#### d. Veracity

Data mentah yang masuk ke database perusahaan biasanya berjumlah sangat banyak dan berantakan. Mengingat data tersebut bisa diperoleh dari berbagai sumber, data belum tentu akurat. Veracity (kebenaran) mengartikan big data harus memiliki kebenaran atau tingkat kepercayaan tertentu agar menghasilkan wawasan yang tepat.

#### e. Value

Kumpulan data yang dimiliki perusahaan tidak akan berguna jika tidak diubah menjadi sesuatu yang bernilai. Value atau nilai mengacu pada pengertian bahwa big data harus memiliki nilai bagi bisnis. Data mentah perlu diubah menjadi sesuatu yang berharga dan memberikan wawasan/informasi penting.

#### 2.1.3. Manfaat Big Data untuk Bisnis

Menurut ringkasan dari Tech Target, berikut beberapa kegunaan big data dalam dunia bisnis:

- Memahami Pelanggan Secara Lebih Mendalam

Perusahaan saat ini mengandalkan data untuk memperoleh pemahaman yang lebih baik tentang pelanggan. Mereka mengumpulkan informasi pelanggan dari berbagai sumber, termasuk media sosial dan cookie.

Data ini membantu perusahaan memahami produk atau layanan yang diminati pengguna sehingga mereka dapat menawarkan produk yang sesuai dengan kebutuhan dan keinginan pelanggan.

- Meningkatkan pemahaman pasar

Big data berperan penting dalam meningkatkan pemahaman pasar. Sebagai contoh, dengan memantau media sosial, perusahaan dapat mengetahui produk atau layanan yang paling banyak dibicarakan, diulas, atau direkomendasikan oleh pengguna. Informasi ini memungkinkan perusahaan untuk menganalisis preferensi dan kecenderungan konsumen terhadap jenis produk tertentu.

- Menyasar Audiens Dengan Lebih Efektif

Saat pengguna mencari produk di mesin pencari atau situs e-commerce, mereka seringkali melihat iklan produk yang relevan di media sosial atau platform iklan Google mereka. Ini adalah hasil dari pemanfaatan big data.

- Perangkat Mencatat Riwayat Pencarian Pengguna Dan Menggunakan Informasi Tersebut Untuk Menampilkan Iklan Produk Serupa

Big data membantu perusahaan menyasar pelanggan dengan lebih efektif karena memungkinkan mereka untuk menyesuaikan iklan dengan audiens yang telah mencari produk serupa sebelumnya.

- Mendorong Inovasi Berbasis Data



Big data tidak memberikan wawasan secara otomatis. Data perlu dianalisis secara teliti dan disusun dengan baik hingga menjadi informasi yang berharga. Dengan big data, perusahaan dapat mengidentifikasi peluang inovasi produk baru yang sesuai dengan kebutuhan dan keinginan pelanggan.

#### 2.1.4. Contoh Penerapan Big Data

##### 2.1.4.1. Bidang Marketing

###### 2.1.4.1.1. Netflix

Pada tahun 2022, Netflix memiliki sekitar 223 juta pelanggan di seluruh dunia. Pertumbuhan jumlah pelanggan yang signifikan ini menyebabkan akumulasi data pelanggan yang semakin besar, yang pada akhirnya menjadi big data. Netflix memanfaatkan big data ini untuk berbagai keperluan, termasuk melacak riwayat tontonan pelanggan dan menganalisis rata-rata waktu menonton. Informasi yang diperoleh dari analisis big data ini digunakan oleh Netflix untuk menyajikan rekomendasi film yang lebih personal dan relevan kepada pengguna, meningkatkan pengalaman menonton mereka.

###### 2.1.4.1.2. McDonald's

McDonald's memanfaatkan big data untuk menciptakan kupon penawaran yang efektif. Proses ini dimulai dengan pengumpulan data dari berbagai sumber, termasuk aplikasi McDonald's, menu digital di restoran, dan informasi pelanggan yang bertransaksi melalui layanan drive-thru.

Dengan menggunakan data yang terkumpul, McDonald's dapat menganalisis kebiasaan pembelian dan preferensi menu dari pelanggan. Informasi ini kemudian digunakan untuk merancang berbagai promosi dan penawaran yang sesuai dengan preferensi konsumen. Dengan demikian, McDonald's dapat meningkatkan efektivitas promosi mereka dan memberikan penawaran yang lebih menarik kepada pelanggan.



Gambar 1. 3 Contoh Pemanfaatan Big Data Untuk Penawaran Mcdonald's

##### 2.4.1.2. Bidang Kesehatan

###### 2.4.1.2.1. Electronic health record

Dalam menghadapi peningkatan jumlah pasien, rumah sakit mengalami kesulitan jika masih menggunakan pencatatan manual dengan kertas untuk menyimpan riwayat data pasien. Kondisi ini mendorong kehadiran Electronic Health Record (EHR) atau catatan kesehatan elektronik.

EHR merupakan sistem pencatatan kesehatan pasien dalam format digital yang menggantikan penggunaan kertas. Data yang terdapat dalam EHR mencakup beragam informasi, seperti riwayat penyakit, alergi, dan riwayat pengobatan pasien.

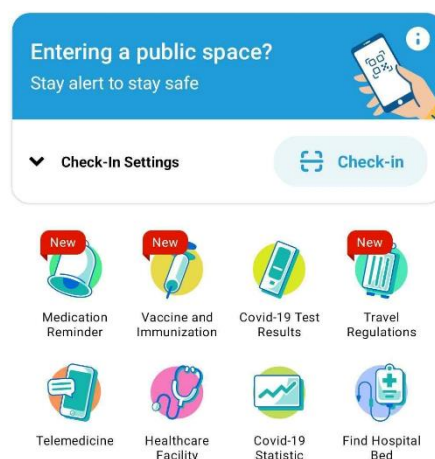
Dengan adopsi format digital, rumah sakit dapat mengumpulkan dan menyimpan semua data pasien dalam bentuk big data yang bermanfaat. Sebagai contoh, dokter memiliki akses langsung untuk melihat data pasien yang sedang ditanganinya. Selain itu, sistem EHR juga dapat memberikan pengingat kepada dokter ketika pasien perlu menjalani pemeriksaan rutin atau kontrol. Dengan demikian, dokter dapat mengirimkan pesan pengingat kepada pasien tentang jadwal kontrol yang telah ditetapkan.

#### 2.4.1.2.2. Disease tracking (dalam upaya penanganan Covid-19)

Selama pandemi Covid-19, pemerintah mengambil langkah-langkah untuk mengendalikan penyebarannya, salah satunya adalah dengan mengembangkan aplikasi Peduli Lindungi yang tersedia untuk diunduh di berbagai jenis smartphone.

Peduli Lindungi merupakan aplikasi yang dirancang untuk membantu pemerintah dalam melakukan pelacakan kontak dan memberikan peringatan kepada pengguna ketika ada warga yang terkonfirmasi positif Covid-19.

Aplikasi ini mengandalkan big data untuk menyediakan berbagai fitur yang bermanfaat, termasuk melihat statistik kasus Covid-19 di wilayah tertentu, menampilkan hasil tes Covid-19, dan memungkinkan pengguna untuk melakukan pemeriksaan kesehatan secara mandiri. Dengan menggunakan teknologi ini, pemerintah berharap dapat lebih efektif dalam mengidentifikasi dan mengendalikan penyebaran virus, serta memberikan informasi yang relevan kepada masyarakat untuk menjaga kesehatan dan keselamatan mereka selama pandemi.



*Gambar 1. 4 Contoh Pemanfaatan Big Data di Aplikasi PeduliLindungi*

Aplikasi PeduliLindungi juga memungkinkan pengguna untuk menunjukkan status vaksinasi mereka. Ketika melakukan perjalanan atau mengunjungi tempat yang memerlukan

bukti vaksinasi, masyarakat dapat dengan mudah menunjukkan sertifikat vaksinasi mereka langsung dari aplikasi ini. Ini memberikan kemudahan bagi pengguna dalam memverifikasi status vaksinasi mereka tanpa harus membawa dokumen fisik, serta membantu dalam memastikan keamanan dan kesehatan di tempat-tempat yang menerapkan kebijakan vaksinasi.

## 2.2. Apache Hadoop

### 2.2.1. Sejarah Hadoop

Hadoop menjadi sebuah titik fokus dalam dunia Big Data. Awalnya terinspirasi dari publikasi makalah Google MapReduce dan Google File System (GFS) pada Oktober 2003. Makalah tersebut memberikan gambaran tentang kompleksitas Big Data yang digunakan oleh Google untuk menampung data dalam skala yang sangat besar. Pada tahun 2005, Doug Cutting dan Mike Cafarella mengembangkan Hadoop saat bekerja di Yahoo, dan nama "Hadoop" sendiri diambil dari mainan gajah berwarna kuning milik anak Doug Cutting.

Hadoop menjadi solusi utama dalam menangani tantangan utama Big Data, yaitu Volume, Velocity, dan Variety. Data yang harus diolah sangat besar, memerlukan akses yang cepat, dan memiliki variasi yang sangat bervariasi sehingga tidak dapat ditangani oleh metode manajemen data konvensional. Hadoop berfungsi sebagai platform yang mampu memproses data dalam jumlah besar secara bersamaan. Dengan menggunakan Hadoop, data berukuran raksasa dapat dialirkan ke server cluster distribusi dan diolah menggunakan aplikasi analisis terdistribusi di setiap cluster.

Konsep Hadoop seperti memiliki beberapa tukang cat yang bekerja secara bersamaan untuk mempersingkat waktu pengecatan satu ruangan kamar rumah. Hadoop bekerja dengan cluster dan terdistribusi, dirancang untuk tetap handal meskipun ada satu atau beberapa server atau kluster yang mengalami kegagalan. Keunggulan lainnya adalah cara kerjanya yang tidak memerlukan transfer data bervolume besar lintas jaringan.

### 2.2.2. Apa Itu Hadoop?

Hadoop merupakan sebuah framework yang diciptakan oleh Google dan Apache Software Foundation, yang secara khusus dirancang untuk mendukung pekerjaan terkait dengan Big Data Analytics. Dalam konteks ini, Hadoop menjadi sebuah alat yang mampu mengatasi berbagai tantangan dalam pengolahan Big Data yang sebelumnya hanya dapat diatasi secara konvensional.

Sebelum adanya Hadoop, pengolahan Big Data seringkali terkendala oleh heterogenitas data, termasuk structured data, semi-structured data, dan unstructured data. Mengingat pentingnya peran Big Data dalam ranah bisnis, diperlukan pendekatan pengolahan yang lebih efisien dan efektif. Keberadaan framework Hadoop memberikan solusi dengan memungkinkan pengolahan data dalam skala yang lebih besar, menyimpan data heterogen, dan mempercepat proses pengolahan data.

Dilansir dari laman AWS, Hadoop diakui sebagai framework open source yang sangat efektif dalam penyimpanan dataset dalam jumlah yang sangat besar. Selain kemampuannya dalam penyimpanan, Hadoop juga mampu memproses data mulai dari ukuran gigabyte hingga

petabyte dengan efisien, menjadikannya sebagai pilihan utama dalam lingkup Big Data Analytics.

### 2.2.3. Cara Kerja Hadoop

Hadoop, sebagai sebuah sistem yang kompleks namun efisien dalam mengelola Big Data, mengoperasikan empat modul utama yang memberikan fondasi bagi fungsionalitasnya, berikut penjelasannya:

- Hadoop Distributed File System (HDFS)

Hadoop Distributed File System (HDFS) berperan sebagai sistem terdistribusi yang beroperasi di berbagai tipe hardware, mulai dari standar hingga low-end. Sistem ini memungkinkan penyimpanan data dalam skala besar dengan efisiensi yang tinggi.

- Yet Another Resource Negotiator (YARN)

Yet Another Resource Negotiator (YARN) bertanggung jawab atas pengaturan dan pemantauan penggunaan sumber daya dan node cluster. YARN memastikan alokasi yang optimal dari sumber daya yang tersedia di lingkungan Hadoop.

- MapReduce

MapReduce menjadi framework utama dalam melakukan komputasi data secara paralel, memungkinkan program untuk menjalankan tugas dengan efisien di seluruh cluster Hadoop.

- Hadoop Common

Hadoop Common adalah penyedia library Java yang mendukung fungsionalitas semua modul Hadoop, memungkinkan integrasi yang lancar antara komponen-komponen tersebut. Dengan proses distribusi dataset ke berbagai mesin dan pemrosesan data secara paralel, Hadoop memanfaatkan HDFS untuk penyimpanan data, sementara MapReduce bertugas dalam memproses data tersebut. YARN kemudian mengatur pembagian tugas di dalam kluster, menciptakan sistem yang terkoordinasi dan efisien dalam pengelolaan dan analisis Big Data.

### 2.2.4. Kelebihan Hadoop

Hadoop menawarkan sejumlah keunggulan yang menjadikannya pilihan utama dalam pengelolaan dan analisis Big Data.

- Fleksibel

Fleksibilitasnya memungkinkan data disimpan dalam berbagai format, baik structured maupun unstructured. Ini memberikan kemampuan bagi pengguna untuk mengakses data dari berbagai sumber dengan tipe yang beragam.

- Upgrade kapasitas

Selanjutnya, kelebihan Hadoop terletak pada kemampuannya untuk mengupgrade kapasitas penyimpanan. Berbeda dengan sistem tradisional yang memiliki batasan pada kapasitas penyimpanan, Hadoop mampu ditingkatkan kapasitasnya karena bekerja secara terdistribusi, memungkinkan pengguna untuk mengelola dan menyimpan data dalam skala yang lebih besar.

- Ketahanan tinggi

Ketahanan tinggi juga menjadi salah satu keunggulan utama Hadoop, terutama melalui komponen HDFS. Dengan ketahanan tinggi ini, risiko kegagalan baik dari segi software maupun hardware dapat diminimalkan. Bahkan jika satu node mengalami kerusakan atau masalah, HDFS tetap mampu menyediakan backup data untuk memastikan kelancaran proses pengolahan data.

Dengan kombinasi fleksibilitas, kapasitas yang dapat diupgrade, dan ketahanan tinggi, Hadoop membuktikan dirinya sebagai solusi yang andal dan efisien dalam mengelola Big Data.

## 2.2.5. Jenis Instalasi Hadoop

Hadoop dapat diinstal dengan berbagai jenis mode, memberikan fleksibilitas kepada data scientist sesuai dengan kebutuhan mereka. Secara prinsip, Hadoop beroperasi pada sistem operasi komputer berbasis Unix atau Linux, meskipun pengguna juga memiliki opsi untuk mengimplementasikannya pada perangkat berbasis Windows, walaupun tidak disarankan.

Berikut adalah berbagai tipe instalasi Hadoop memberikan variasi dalam metode dan proses kerja:

- Standalone mode

Standalone mode merupakan bentuk paling sederhana yang berjalan pada satu node atau sistem. Mode ini menggunakan proses JVM tunggal untuk mensimulasikan sistem terdistribusi dan menggunakan sistem file lokal untuk penyimpanan. Meskipun tidak mendukung HDFS dan YARN, Standalone mode sering digunakan untuk menguji pekerjaan yang berorientasi pada program MapReduce sebelum dijalankan di sebuah cluster.

- Pseudo-distributed mode

Pseudo-distributed mode, menggabungkan kualitas Standalone dan fully-distributed mode. Meskipun berjalan pada satu node, terdapat dua proses JVM untuk mensimulasikan dua node, satu sebagai master dan satu lagi untuk penopang kerja. Mode ini biasanya digunakan untuk pengujian lingkungan yang lengkap.

- Fully-distributed mode

Terakhir, Fully-distributed mode merupakan jenis instalasi Hadoop yang paling penting, dirancang untuk environment produksi yang berjalan pada sekelompok mesin terdistribusi nyata. Dalam mode ini, beberapa node menjalankan Daemon Master seperti Namenode dan Resource Manager, sementara sisa nodenya menjalankan Daemon Slave seperti DataNode dan Node Manager. Jenis instalasi ini memainkan peran krusial dalam mengoptimalkan kinerja Hadoop dalam mengelola Big Data.

## 2.2.6. Jenis Hadoop

Hadoop, sebagai platform perangkat lunak open-source, menawarkan berbagai jenis distribusi yang dapat disesuaikan dengan kebutuhan pengguna. Beberapa jenis Hadoop yang umum digunakan antara lain:

- Apache Hadoop

Apache Hadoop merupakan distribusi resmi dari platform Hadoop yang dikembangkan oleh Apache Software Foundation. Apache Hadoop menyediakan komponen utama seperti HDFS dan MapReduce, serta alat tambahan seperti Hive, Pig, dan Spark untuk analisis data kompleks.

- Cloudera

Cloudera menjadi salah satu distributor Hadoop yang terpopuler, menyediakan distribusi Hadoop yang sudah dikonfigurasi dan mudah digunakan, serta alat tambahan seperti Cloudera Manager, Impala, dan Search.

- Hortonworks

Hortonworks, sebagai perusahaan yang berfokus pada pengembangan platform Hadoop, menyediakan distribusi Hadoop lengkap dan layanan konsultasi untuk implementasi di dalam organisasi.

- MapR

MapR, dengan fokus pada kinerja dan keamanan data, menggunakan teknologi file sistem MapR-FS yang lebih cepat dan aman daripada HDFS, serta menyediakan alat tambahan seperti MapR-DB dan MapR Streams untuk analisis data real-time.

- IBM Open Platform with Apache Hadoop

IBM Open Platform with Apache Hadoop, dikembangkan oleh IBM, menyediakan alat tambahan seperti BigInsights dan Analytics for Apache Hadoop.

Dengan berbagai pilihan ini, pengguna dapat memilih jenis Hadoop yang paling sesuai dengan kebutuhan dan kondisi organisasi mereka.

#### 2.2.7. Manfaat Hadoop

Hadoop membawa berbagai manfaat yang signifikan, khususnya dalam mengelola dan menganalisis data yang besar dan kompleks. Beberapa manfaat utama Hadoop antara lain:

- **Skalabilitas:** Skalabilitasnya yang dirancang untuk mengolah data dalam jumlah yang sangat besar, dengan ribuan node di dalam cluster, memungkinkan organisasi untuk menyimpan dan memproses data dalam skala yang lebih besar dibandingkan dengan solusi tradisional.
- **Biaya:** Aspek biaya yang terjangkau karena Hadoop adalah platform open-source yang gratis digunakan, membantu organisasi menghemat biaya pengolahan data.
- **Kecepatan:** Kecepatan pemrosesan data secara paralel memungkinkan penggunaan Hadoop untuk memproses data lebih cepat dan membuat keputusan secara real-time.
- **Analisis data:** Hadoop juga menyediakan berbagai tools analisis data seperti Hive, Pig, dan Spark yang mendukung analisis data yang kompleks, termasuk data mining, prediksi, dan machine learning.
- **Fleksibilitas:** Hadoop memungkinkan penggunaannya dalam berbagai environment, seperti cloud, on-premise, atau hybrid, memberikan organisasi pilihan lingkungan yang sesuai dengan kebutuhan mereka.
- **Keamanan:** Hadoop juga mengutamakan keamanan dengan menyediakan mekanisme autentikasi, otorisasi, dan enkripsi untuk melindungi data dari akses yang tidak sah.
- **Ketersediaan:** Arsitektur terdistribusi dan redundan yang digunakan oleh Hadoop membantu meningkatkan ketersediaan data dan mengurangi resiko kehilangan data.

Dengan beragam manfaat ini, Hadoop menjadi solusi yang kuat bagi organisasi dalam mengelola dan menganalisis data yang besar dan kompleks.

#### 2.2.8. Contoh Penggunaan Hadoop

Hadoop dapat di aplikasikan dalam berbagai industri yang berbeda, memperkuat kemampuan organisasi dalam mengelola, memproses, dan menganalisis data dalam skala besar dan kompleks. Contoh penggunaan Hadoop di berbagai industri :

- **Industri keuangan**

Di industri keuangan, Hadoop memainkan peran penting dalam mengelola data historis dan transaksi keuangan. Dengan kemampuannya yang luas, Hadoop membantu perusahaan keuangan melakukan analisis risiko kredit dengan lebih akurat, mendeteksi kecurangan secara efisien, serta mengelola risiko dengan lebih efektif. Dengan memproses dan menganalisis data transaksi yang besar dan kompleks, Hadoop memungkinkan perusahaan keuangan untuk mengidentifikasi pola-pola yang mungkin tersembunyi, memberikan wawasan yang mendalam, dan mengambil keputusan yang lebih cerdas dalam mengelola portofolio mereka.

- E-commerce

Dalam industri e-commerce, Hadoop memiliki peran yang signifikan dalam memproses data pelanggan, termasuk riwayat pembelian, preferensi produk, dan data geografis. Dengan menggunakan Hadoop, perusahaan e-commerce dapat menganalisis pola pembelian dan perilaku pelanggan dengan lebih mendalam. Hal ini memungkinkan mereka untuk menyusun strategi pemasaran yang lebih efektif dan membuat rekomendasi produk yang lebih personal kepada pelanggan. Dengan memahami preferensi dan kebiasaan belanja pelanggan, perusahaan e-commerce dapat meningkatkan pengalaman belanja online dan meningkatkan loyalitas pelanggan.

- Telekomunikasi

Hadoop memiliki peran yang krusial dalam industri telekomunikasi dengan kemampuannya untuk memproses data yang dihasilkan oleh jutaan pelanggan. Data yang diproses meliputi informasi tentang penggunaan layanan, kondisi jaringan, dan data geografis. Dengan menggunakan Hadoop, perusahaan telekomunikasi dapat menganalisis tren penggunaan layanan, mengidentifikasi pola kebutuhan pelanggan, serta mengoptimalkan kinerja jaringan mereka. Informasi yang diperoleh dari analisis data ini memungkinkan perusahaan telekomunikasi untuk meningkatkan kualitas layanan yang mereka tawarkan dan membuat keputusan yang lebih baik dalam pengembangan produk dan strategi bisnis.

- Kesehatan

Hadoop memainkan peran penting dalam industri kesehatan dengan kemampuannya untuk memproses beragam data medis, termasuk data pasien, data klinis, dan data penelitian. Dengan menggunakan Hadoop, rumah sakit dan organisasi kesehatan dapat menganalisis data tersebut untuk mengidentifikasi tren kesehatan, mendiagnosis penyakit lebih akurat, dan merancang rencana pengobatan yang lebih efektif. Informasi yang diperoleh dari analisis data medis ini dapat membantu penyedia layanan kesehatan dalam meningkatkan pelayanan kepada pasien, mengoptimalkan efisiensi operasional, dan melakukan penelitian lebih lanjut untuk kemajuan kedokteran.



- Pemerintah

Pemanfaatan Hadoop dalam konteks pemerintahan memberikan kontribusi besar dalam pengelolaan data yang beragam, termasuk data demografi, keamanan, dan kesehatan masyarakat. Dengan Hadoop, pemerintah dapat mengintegrasikan data dari berbagai sumber untuk mendapatkan pemahaman yang lebih baik tentang dinamika populasi, keamanan nasional, dan status kesehatan masyarakat. Analisis yang dilakukan atas data tersebut memungkinkan pemerintah untuk membuat keputusan yang lebih baik dan efektif dalam merencanakan kebijakan publik, mengalokasikan sumber daya, dan menangani isu-isu kritis yang berkaitan dengan kesejahteraan masyarakat secara keseluruhan.

Hadoop dapat membantu organisasi dalam mengelola, memproses, dan menganalisis data dalam skala besar dan kompleks, sehingga membantu dalam pengambilan keputusan yang lebih baik dan efektif.

Dengan berbagai keunggulan dan kemampuannya dalam mengatasi tantangan Big Data, Hadoop menjadi salah satu tools yang sangat penting dalam ranah Big Data Analytics. Dengan terus berkembangnya teknologi dan pertumbuhan jumlah data, Hadoop tetap menjadi salah satu solusi utama bagi perusahaan-perusahaan yang berusaha menggali wawasan dari data dalam skala besar.

## 2.3. Instalasi dan konfigurasi Hadoop

Hadoop adalah platform open source berbasis Java yang digunakan untuk mendukung aplikasi yang berjalan pada big data. Berikut adalah langkah-langkah umum untuk menginstal dan mengonfigurasi Hadoop 3.3.6 di Windows 10:

### 2.3.1. Prasyarat

- a. Pastikan Anda telah menginstal Java versi 8. Semua versi Hadoop hanya mendukung Java versi 8.
- b. Unduh dan instal Java Development Kit 8 (JDK 8) dari situs Oracle.
- c. Setelah menginstal JDK 8, atur Environment Variables untuk Java.

### 2.3.2. Proses Instalasi Hadoop

- a. Unduh Hadoop versi 3.3.6. dari situs resmi Apache Hadoop.
- b. Pilih opsi download tar.gz.
- c. Salin file Hadoop ke drive C dan ekstrak file tersebut.
- d. Ganti nama folder hasil ekstraksi dari “hadoop-3.3.6.” menjadi “hadoop”.

### 2.3.3. Konfigurasi Hadoop

- a. Setelah menginstal Hadoop, konfigurasi file konfigurasi seperti 'core-site.xml', 'hdfs-site.xml', dan 'mapred-site.xml'.
- b. Atur variabel lingkungan seperti HADOOP\_HOME dan PATH untuk mengakses perintah Hadoop dari mana saja.

### 2.3.4. Mulai Hadoop

- a. Buka terminal dan jalankan perintah start-all.sh untuk memulai semua daemon Hadoop.
- b. Buka web interface Hadoop di <http://localhost:50070> untuk memeriksa status cluster.

## 2.4. Apache Spark

### 2.4.1. Apa itu Apache Spark?

Apache Spark adalah sebuah platform data processing terdistribusi open-source yang dirancang untuk pemrosesan dan analisis data skala besar dengan kecepatan tinggi.

Platform ini menonjol karena kemampuannya dalam in-memory processing, memungkinkannya melakukan operasi data dengan kecepatan jauh lebih tinggi dibandingkan dengan sistem pemrosesan batch tradisional lain. Dengan kemampuan in-memory computing-nya, Spark mampu mengurangi jumlah pembacaan dan penulisan ke disk, yang secara signifikan meningkatkan kecepatan pemrosesan data dibandingkan dengan sistem yang bergantung pada operasi disk. Hal ini menjadikan Spark efektif untuk analisis data yang memerlukan respons cepat, seperti analisis interaktif dan machine learning.

Arsitektur Spark berpusat pada konsep Resilient Distributed Datasets (RDD), yaitu kumpulan data terdistribusi yang dapat dioperasikan secara paralel. RDD ini dirancang untuk efisiensi, toleransi kesalahan, dan kemudahan penggunaan dalam pemrosesan data terdistribusi. Spark menyediakan API yang mumpuni untuk Scala, Java, Python, dan R, sehingga memudahkan pengembangan aplikasi.

Selain itu, Spark mendukung berbagai jenis workload, termasuk batch processing, streaming data, kueri interaktif, machine learning, dan pemrosesan grafik.

### 2.4.2. Sejarah Apache Spark

Apache Spark berasal dari sebuah inisiatif di AMPLab, University of California, Berkeley pada tahun 2009. Tujuan utama proyek ini adalah untuk mengatasi batasan yang ada pada model pemrograman MapReduce, yang digunakan untuk melakukan pemrosesan dataset besar secara terdistribusi dan paralel. MapReduce, yang dikembangkan oleh Google, menghadapi tantangan dalam proses multi-langkah yang memerlukan banyak pembacaan dan penulisan data ke disk, sehingga menyebabkan keterlambatan dalam eksekusi.

Spark dikembangkan untuk meningkatkan efisiensi serta mempermudah penggunaan MapReduce. Dengan melakukan pemrosesan in-memory dan mengurangi jumlah langkah dalam sebuah pekerjaan, Spark mampu menjalankan tugas lebih cepat. Selain itu, Spark

memanfaatkan cache in-memory untuk mempercepat algoritma pembelajaran mesin dengan melakukan pemanggilan fungsi yang sama berulang kali pada dataset yang sama.

Pada tahun 2013, Spark memasuki tahap inkubasi di Apache Software Foundation, dan pada awal 2014, proyek ini naik ke status proyek tingkat atas di dalam Foundation. Saat ini, Spark telah menjadi proyek yang sangat aktif di bawah naungan Apache Software Foundation, didukung oleh komunitas yang terdiri dari kontributor individu serta dukungan dari perusahaan-perusahaan besar seperti Databricks, IBM, dan Huawei dari China.

#### 2.4.3. Kegunaan Apache Spark

Menurut informasi dari Amazon Web Services dan Databricks, Apache Spark memiliki beberapa kegunaan utama yang meliputi:

1. Pemrosesan data real-time

Spark mampu mengelola data streaming secara real-time, memungkinkan pengolahan data yang masuk secara langsung dan kontinu, seperti data dari sensor atau transaksi keuangan.

2. Analitik interaktif dan query

Spark menyediakan dukungan untuk query interaktif yang memungkinkan analisis untuk menjalankan analisis dan mendapatkan hasil dengan cepat. Fitur ini sangat berguna untuk analisis bisnis dan pengambilan keputusan berbasis data.

3. Machine learning

Dengan kemampuan dalam machine learning, Spark memungkinkan pengembangan model yang dapat belajar dari data besar, berguna dalam berbagai aplikasi seperti rekomendasi produk, deteksi penipuan, dan lainnya.

4. Pemrosesan grafik

Spark juga mendukung pemrosesan grafik, memungkinkan analisis untuk menganalisis dan memanipulasi data yang terstruktur sebagai grafik, seperti social network atau jaringan komunikasi.

5. Dukungan bahasa pemrograman

Spark mendukung berbagai bahasa pemrograman termasuk Java, Scala, Python, dan R, sehingga memudahkan integrasinya ke dalam berbagai aplikasi dan sistem.

6. Analitik lanjutan

Spark juga mendukung SQL queries, memungkinkan analisis untuk melakukan query dan analisis data dengan cara yang lebih familiar bagi mereka yang sudah terbiasa dengan SQL.

#### 2.4.4. Contoh Pemanfaatan Apache Spark

Berikut beberapa contoh pemanfaatan Apache Spark dalam berbagai skenario:

1. Analisis sentimen media sosial

Spark dapat digunakan untuk mengolah dan menganalisis data media sosial secara real-time guna menentukan sentimen umum terhadap suatu topik atau brand. Sebagai contoh, perusahaan dapat menggunakan Spark untuk menganalisis tweet tentang produk mereka guna memahami persepsi pelanggan terhadap produk tersebut.

2. Rekomendasi produk e-commerce

Dengan memanfaatkan algoritma machine learning yang tersedia di MLlib, Spark dapat digunakan untuk mengembangkan sistem rekomendasi yang mempersonalisasi pengalaman belanja pelanggan berdasarkan perilaku pembelian dan preferensi masing-masing.

3. Pemrosesan log data

Spark cocok untuk menganalisis log data dari web server atau aplikasi guna memantau pola traffic, aktivitas pengguna, atau mendeteksi kegiatan mencurigakan seperti serangan cyber.

4. Analisis data keuangan

Spark juga berguna untuk analisis data keuangan skala besar, seperti deteksi penipuan dalam transaksi kartu kredit atau analisis pasar saham secara real-time.

5. Pengolahan data IoT

Dalam konteks Internet of Things (IoT), Spark dapat digunakan untuk mengolah data yang dihasilkan oleh perangkat IoT, seperti sensor, guna melakukan pemantauan kondisi secara real-time atau analisis prediktif.

#### 2.4.5. Perbedaan Apache Hadoop dan Apache Spark

Apache Spark adalah sebuah sistem pemrosesan data terdistribusi open-source yang ditujukan untuk melakukan pemrosesan data dalam skala besar dan analitik dengan cepat. Spark memiliki kemampuan untuk melakukan pemrosesan in-memory, yang memungkinkannya untuk menjalankan operasi data dengan kecepatan tinggi jika dibandingkan dengan sistem tradisional batch processing.

Di sisi lain, Hadoop merupakan sebuah framework open-source yang digunakan untuk penyimpanan dan pemrosesan data dalam skala besar. Hadoop lebih difokuskan pada pemrosesan data dalam batch yang efisien dan mampu menangani berbagai jenis data, mulai dari yang terstruktur hingga tidak terstruktur.

Secara singkat, perbedaan utama antara Apache Spark dan Hadoop terletak pada pendekatan yang digunakan dalam pemrosesan data. Spark lebih dioptimalkan untuk melakukan pemrosesan in-memory dan analitik dengan cepat, sementara Hadoop berorientasi pada pemrosesan data dalam batch yang efisien dan penyimpanan data yang terdistribusi.

Aspek	Apache Spark	Hadoop
<b>Arsitektur</b>	Tidak memiliki sistem file asli. Sering dijalankan di atas HDFS atau menggunakan Amazon S3/Redshift.	Memiliki sistem file asli HDFS yang membagi dan menyimpan data dalam blok-blok kecil.
<b>Kinerja</b>	Proses data secara real-time dengan menyalin data ke RAM sebelum pemrosesan. Lebih cepat dan efisien.	Proses data dalam batch dan mungkin lebih lambat karena membaca dan menulis data ke penyimpanan eksternal.
<b>Machine Learning</b>	Menyediakan MLlib untuk analisis regresi, klasifikasi, dan tugas machine learning lainnya.	Tidak memiliki perpustakaan machine learning bawaan. Dapat diintegrasikan dengan perangkat lunak lain untuk machine learning.
<b>Keamanan</b>	Perlindungan keamanan terbatas. Perlu mengaktifkan fitur keamanan dan memastikan lingkungan yang aman.	Fitur keamanan yang kuat dengan enkripsi dan kontrol akses untuk melindungi data.
<b>Skalabilitas</b>	Skalabilitas memerlukan investasi lebih pada RAM, yang bisa meningkatkan biaya.	Lebih mudah dan murah untuk diskalakan dengan menambahkan node tambahan.
<b>Biaya</b>	Lebih mahal karena menggunakan RAM untuk pemrosesan in-memory.	Lebih terjangkau karena menggunakan hard disk untuk penyimpanan dan pemrosesan data.

*Gambar 1. 5 Perbedaan Apache Hadoop vs Apache Spark*

## 2.5. Kompatibilitas Apache Hadoop dan Apache Spark dengan Windows 10

Hadoop versi 3.3.6 memiliki kompatibilitas dengan Windows 10 untuk penggunaan dalam pengembangan dan pengujian. Meskipun Hadoop awalnya didesain untuk berjalan pada sistem yang mirip dengan Unix, seperti Linux, keberadaan versi Hadoop yang dapat diinstal di lingkungan Windows memungkinkan pengguna Windows untuk memanfaatkan kekuatan dan fitur platform ini tanpa harus beralih ke sistem operasi lain. Ini memperluas fleksibilitas dalam pengembangan dan pengujian aplikasi berbasis big data, memungkinkan pengguna Windows untuk memanfaatkan ekosistem Hadoop tanpa hambatan platform. Dengan demikian, pengguna Windows dapat mengakses dan memanfaatkan kekuatan Hadoop untuk mengelola dan menganalisis data besar sesuai kebutuhan mereka, bahkan jika lingkungan kerja mereka berjalan di atas Windows 10.

Kompatibilitas Apache Spark dengan Windows 10 telah meningkat seiring waktu, namun masih memerlukan beberapa pertimbangan khusus. Meskipun Apache Spark secara teknis dapat dijalankan di Windows 10, proses instalasinya mungkin lebih rumit dibandingkan dengan sistem operasi lain seperti Linux atau macOS. Ini terutama disebabkan oleh perbedaan dalam lingkungan pengembangan dan dependensi perangkat lunak antara Windows dan sistem operasi lainnya. Selain itu, sebagian besar dokumentasi dan tutorial yang tersedia untuk Apache Spark ditujukan untuk lingkungan Linux, sehingga pengguna Windows 10 mungkin perlu menyesuaikan langkah-langkah instalasi dan konfigurasi. Meskipun ada paket binari yang dioptimalkan untuk Windows, dukungan dan pembaruan mungkin tidak sekomprehensif seperti versi Linux. Selain itu, perlu memastikan bahwa versi Java yang dipasang di Windows 10 kompatibel dengan versi Spark yang akan digunakan, dan pengaturan lingkungan seperti

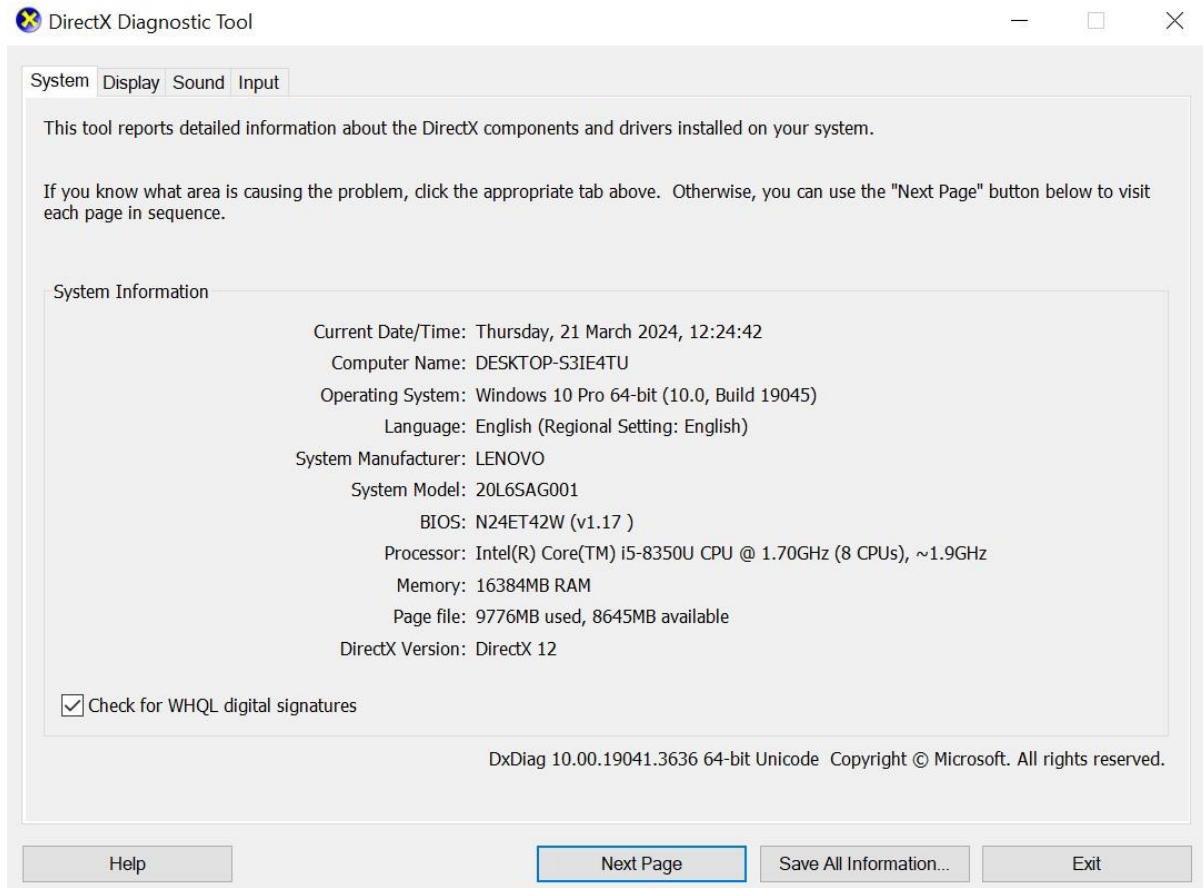
PATH dan variabel lainnya juga perlu diperhatikan dengan cermat agar Spark dapat diakses dengan benar. Meskipun Apache Spark dapat dijalankan di Windows 10, penggunaan platform tersebut masih lebih umum di lingkungan pengembangan berbasis Linux, karena alasan kestabilan, kinerja, dan dukungan komunitas yang lebih luas di platform tersebut. Jika memungkinkan, penggunaan Linux atau menggunakan mesin virtual Linux di Windows 10 mungkin menjadi pilihan yang lebih mudah untuk pengembangan dan pengujian Apache Spark.

## BAB III

### METODE PRAKTIKUM

#### 3.1. Perangkat Praktikum

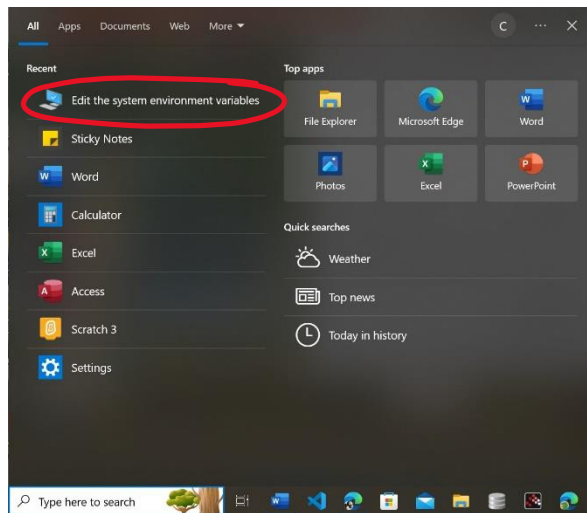
Saya menggunakan perangkat pribadi berupa laptop dengan spesifikasi seperti berikut:



Gambar 1. 6 Spesifikasi Perangkat Komputer yang digunakan untuk Praktikum

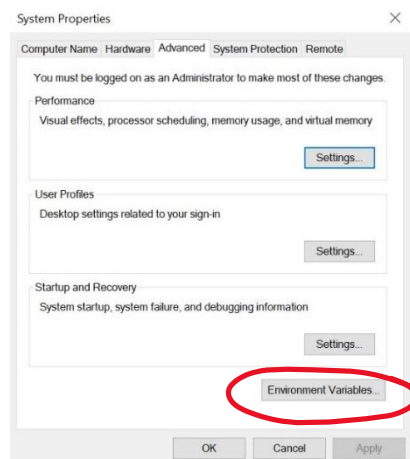
#### 3.2. Prosedur Kerja Praktikum

1. Menyiapkan perangkat praktikum seperti komputer, dan jaringan internet.
2. Sambungkan perangkat dengan jaringan internet.
3. Prasyarat
  - 3.1. Unduh dan instal Java Development Kit 8 (JDK 8) dari situs Oracle.
    - 3.1.1. Kita akan membuat Java environment.
      - 3.1.1.1. Pilih Menu 'Start'
      - 3.1.1.2. Pilih Menu 'Edit the system environment variabels'



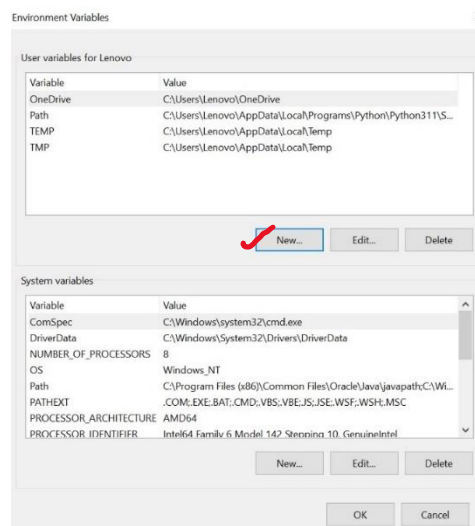
Gambar 1. 7 Menu Start Windows

### 3.1.1.3. Pilih 'Environment Variabels'



Gambar 1. 8 Tampilan System Properties

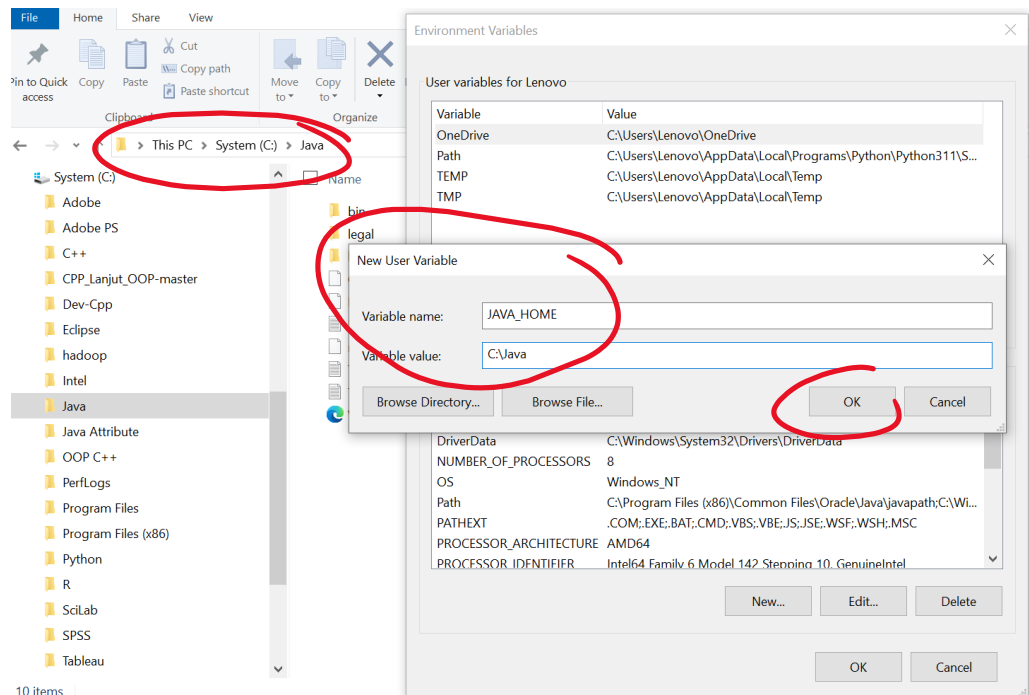
### 3.1.1.4. Pilih 'New' di User Variabels for Lenovo.



Gambar 1. 9 Tampilan Environment Variables

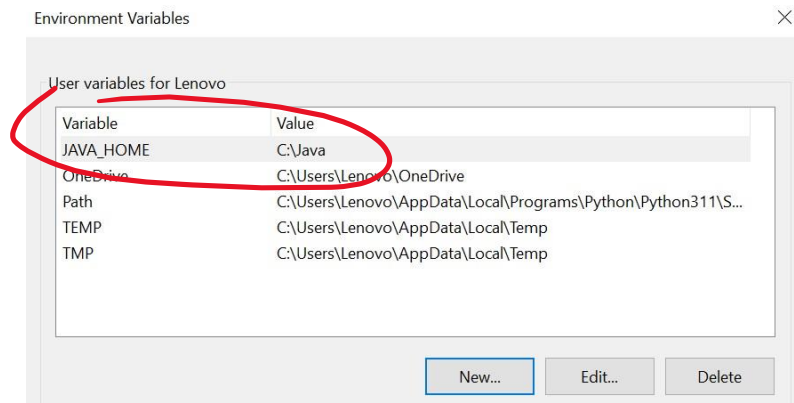


3.1.1.5. Isi kolom Variabel Name dengan “JAVA\_HOME” dan Variabel Value dengan alamat direktori tempat kita menyimpan Java di penyimpanan disk C yaitu, “C:\Java”. Jika sudah klik “Ok”.



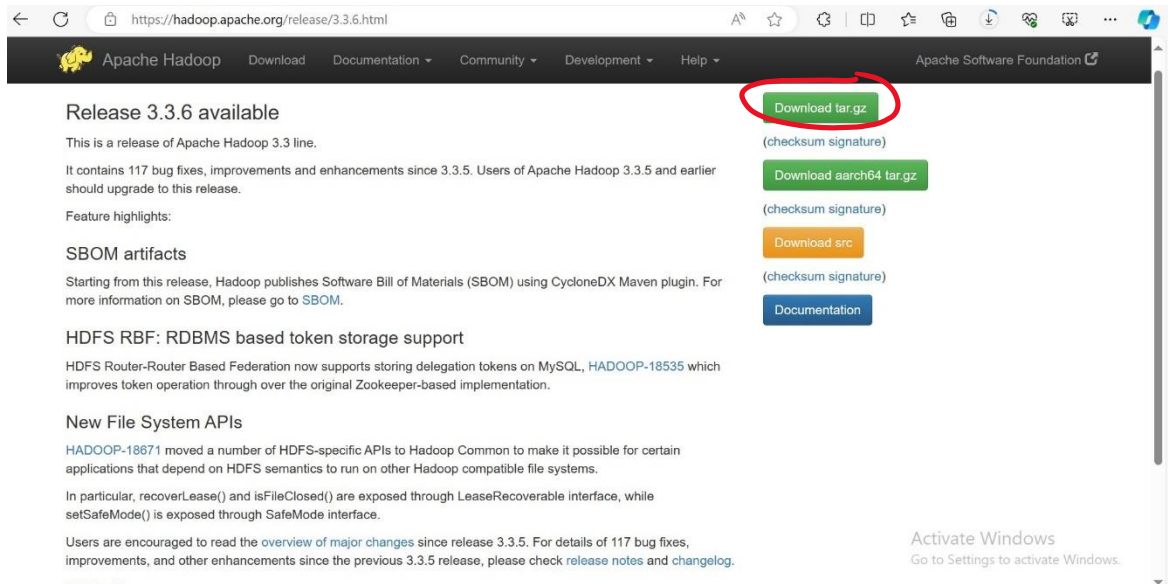
Gambar 1. 10 Tampilan New User Variables untuk Java

Setelah itu ‘JAVA\_HOME’ akan muncul pada User Variables for Lenovo di halaman Environment Variabel.



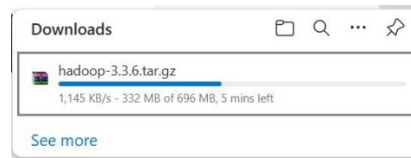
Gambar 1. 11 Tampilan Environment Variables untuk JAVA\_HOME

4. Setelah perangkat tersambung dengan jaringan internet, buka Google Search Engine.
5. Penginstalan Hadoop
  - 5.1. Buka halaman website [Apache Hadoop](#).
  - 5.2. Cari menu download Apache Hadoop versi 3.3.6.



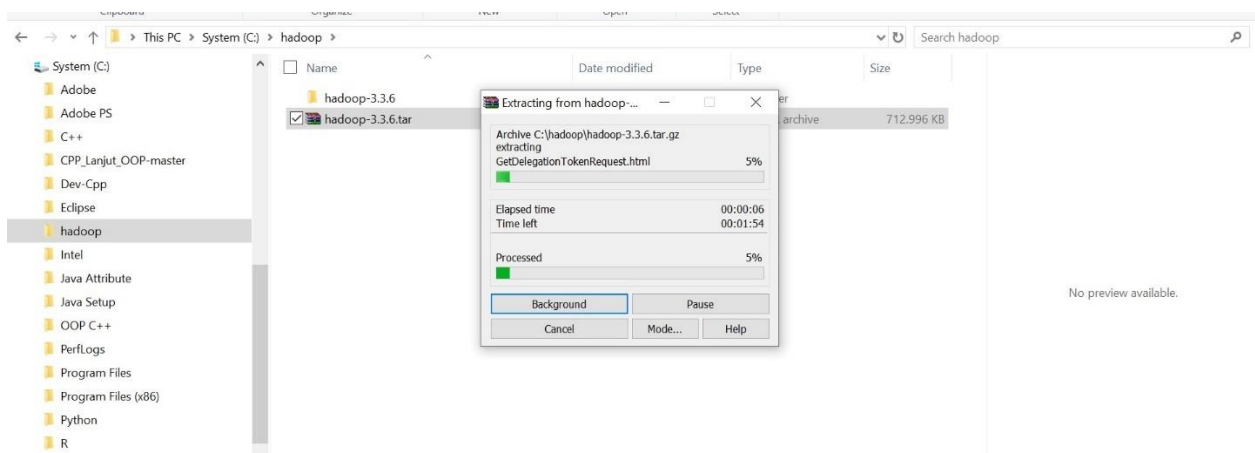
Gambar 1. 12 Tampilan Website Apache Hadoop

### 5.3. Pilih opsi download tar.gz.



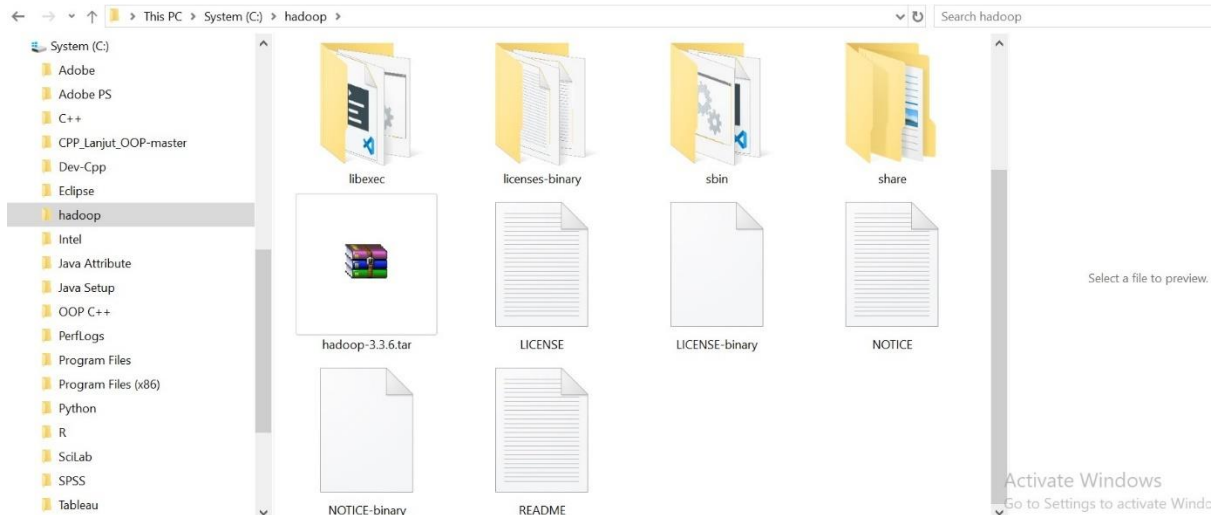
Gambar 1. 13 Proses Downloading Apache Hadoop

### 5.4. Salin file Hadoop ke drive C dan ekstrak file tersebut.



Gambar 1. 14 Proses Extracting Apache Hadoop

### 5.5. Ganti nama folder hasil ekstraksi dari "hadoop-3.3.6." menjadi "hadoop".

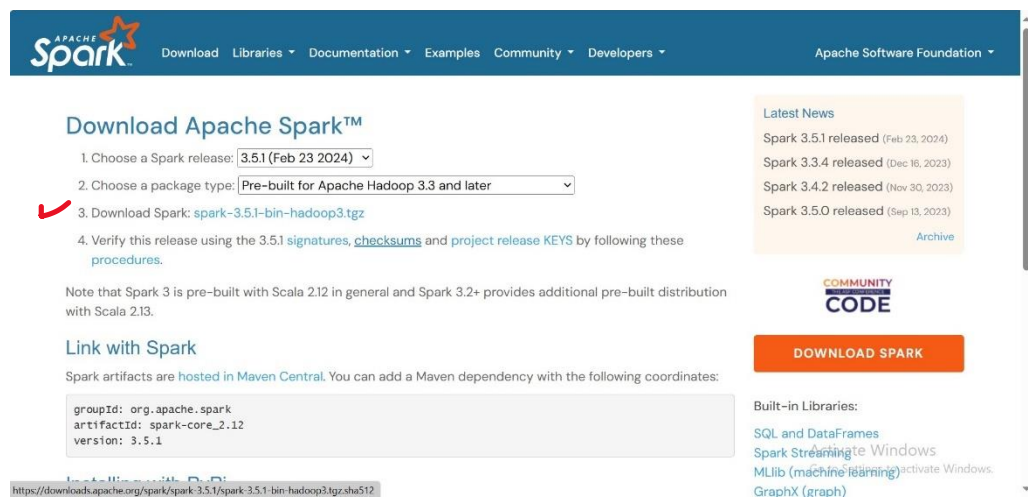


Gambar 1. 15 Ubah Nama Folder Extract Hadoop

## 6. Menginstall Apache Spark

6.1. Buka halaman [Downloads | Apache Spark](#)

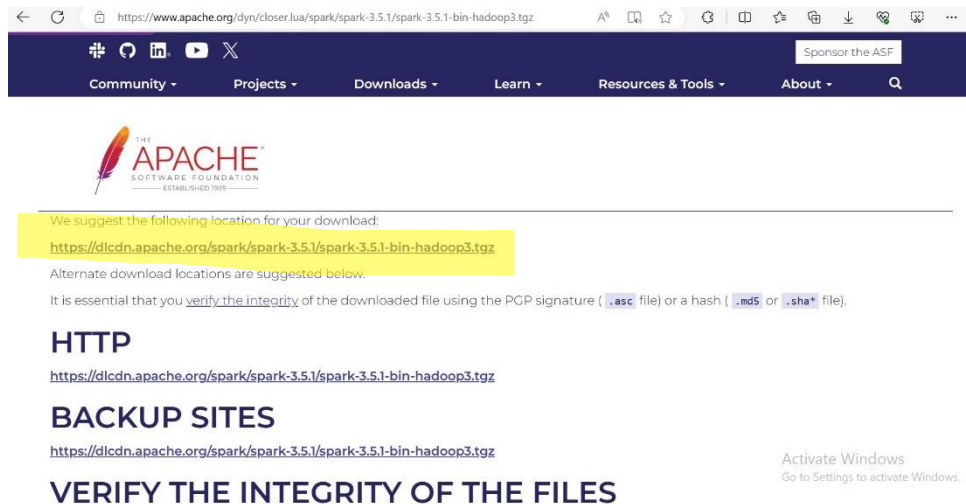
6.2. Pilih nomor 3. Download Spark: spark-3.5.1-bin-hadoop3.tgz



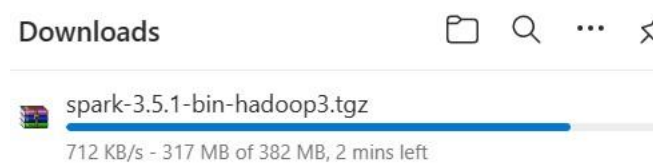
Gambar 1. 16 Tampilan Website Apache Spark

6.3. Pilih *We suggest the following location for your download:*

<https://d1cdn.apache.org/spark/spark-3.5.1/spark-3.5.1-bin-hadoop3.tgz>

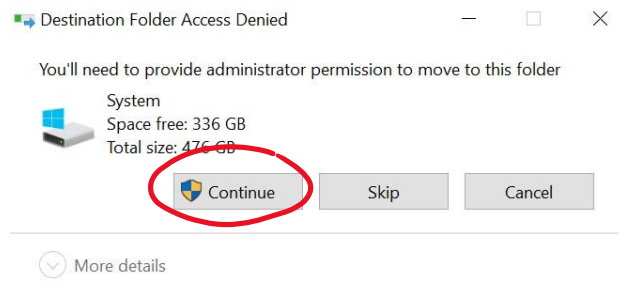


Gambar 1. 17 Tampilan Website Channel untuk Mengunduh Apache Spark

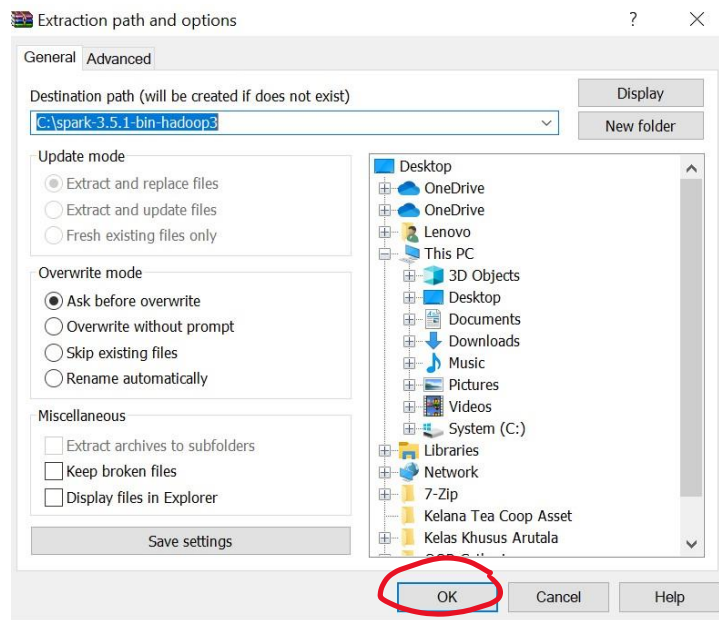


Gambar 1. 18 . Downloading Apache Spark

6.4. Salin file Apache Spark ke drive C dan ekstrak file tersebut.

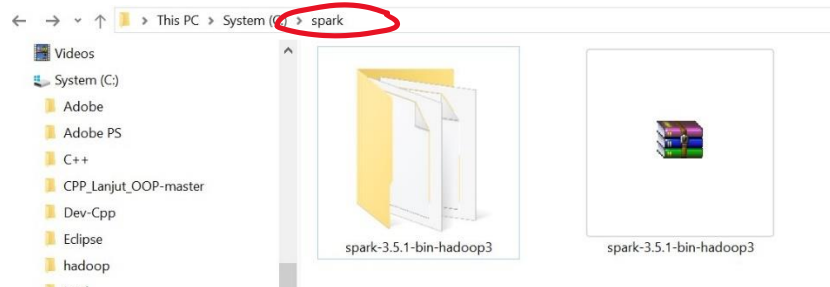


Gambar 1. 19 Peringatan Pemindahan Saat Akan Mengekstrak Folder Apache Spark di Penyimpanan Lokal Disk C



Gambar 1. 20 Extracting Folder Apache Spark

6.5. Ganti nama folder hasil ekstraksi dari “spark-3.5.1-bin-hadoop3” menjadi “spark”.



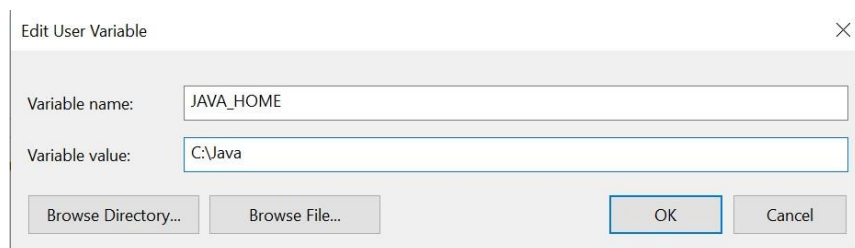
Gambar 1. 21 Mengganti Nama Folder Ekstraksi Apache Spark

## 7. Konfigurasi Java, Hadoop, Spark

### 7.1. Mengatur Environment Variables untuk Java

7.1.1. Pada User Variables for Lenovo pilih ‘New’.

7.1.2. Isi kolom Variabel Name dengan “JAVA\_HOME” dan Variabel Value dengan Java direktori tempat kita menyimpan Java di penyimpanan disk C yaitu, “C:\Java”. Jika sudah klik “Ok”.

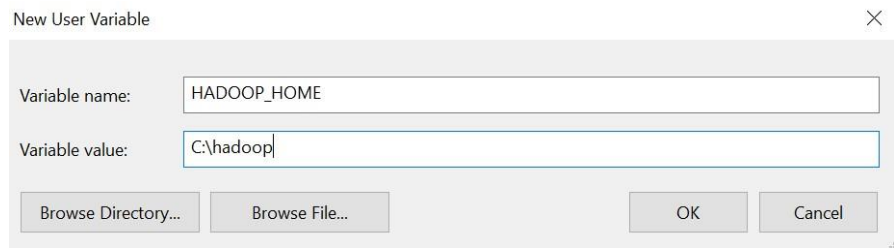


Gambar 1. 22 Menyunting User Variabel dengan JAVA\_HOME

### 7.2. Mengatur Environment Variables untuk Hadoop

7.2.1. Pada User Variables for Lenovo pilih ‘New’.

- 7.2.2. Isi kolom Variabel Name dengan “HADOOP\_HOME” dan Variabel Value dengan hadoop direktori tempat kita menyimpan hadoop di penyimpanan disk C yaitu, “C:\hadoop”. Jika sudah klik “Ok”.



New User Variable

Variable name: HADOOP\_HOME

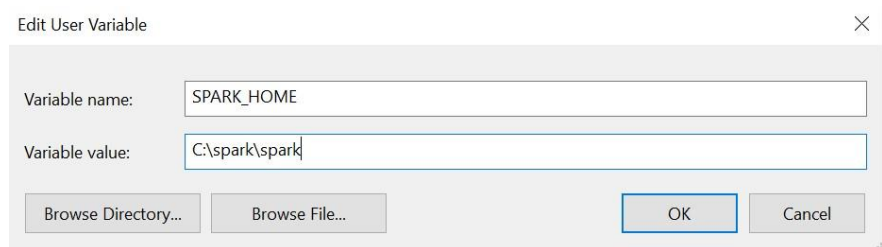
Variable value: C:\hadoop

Browse Directory... Browse File... OK Cancel

*Gambar 1. 23 Menyunting User Variabel dengan HADOOP\_HOME*

### 7.3. Mengatur Environment Variables untuk Spark

- 7.3.1. Pada User Variabels for Lenovo pilih ‘New’.
- 7.3.2. Isi kolom Variabel Name dengan “SPARK\_HOME” dan Variabel Value dengan spark direktori tempat kita menyimpan spark di penyimpanan disk C yaitu, “C:\spark\spark”. Jika sudah klik “Ok”.



Edit User Variable

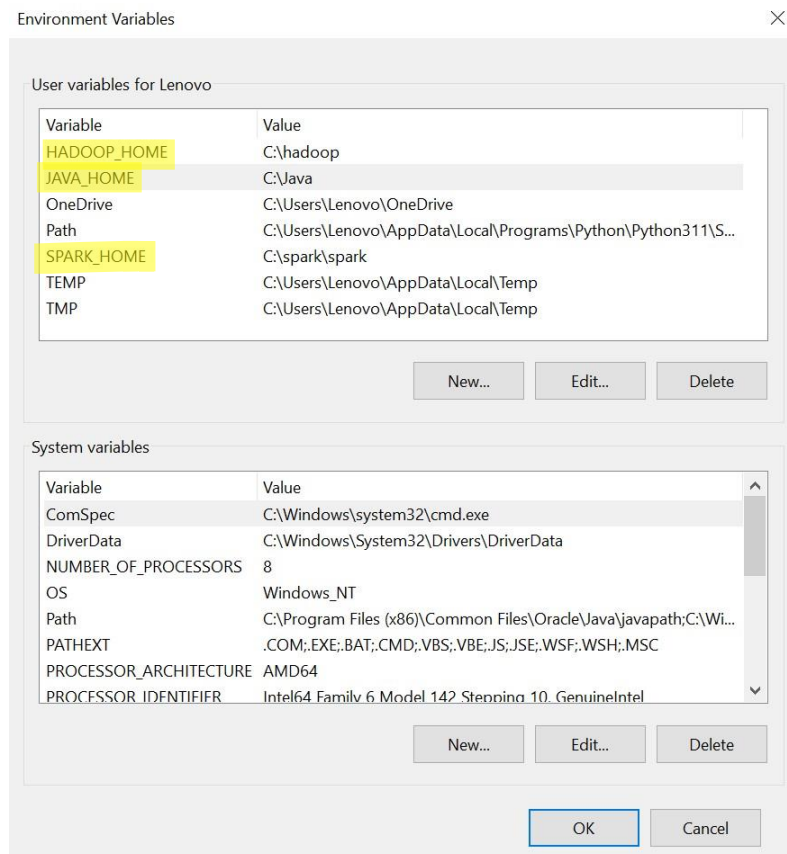
Variable name: SPARK\_HOME

Variable value: C:\spark\spark

Browse Directory... Browse File... OK Cancel

*Gambar 1. 24 Menyunting User Variabel dengan SPARK\_HOME*

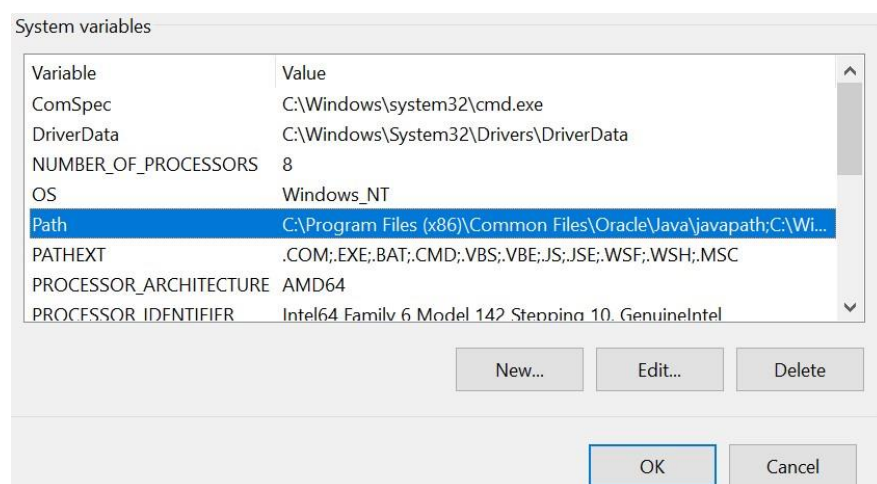
- 7.3.3. Setelah itu ‘JAVA\_HOME’, ‘HADOOP\_HOME’ dan ‘SPARK\_HOME’ akan muncul pada User Variabels for Lenovo di halaman Environment Variabel seperti ini:



Gambar 1. 25 Daftar User Variabel untuk Java, Hadoop dan Spark

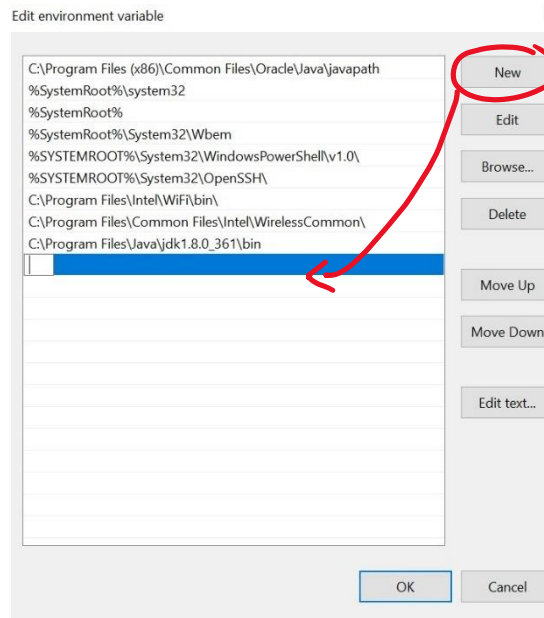
## 7.4. Buat PATH untuk Hadoop, Spark, dan Java

### 7.4.1. Pilih 'Path' pada 'System Variabels'.



Gambar 1. 26 Seleksi Menu Path

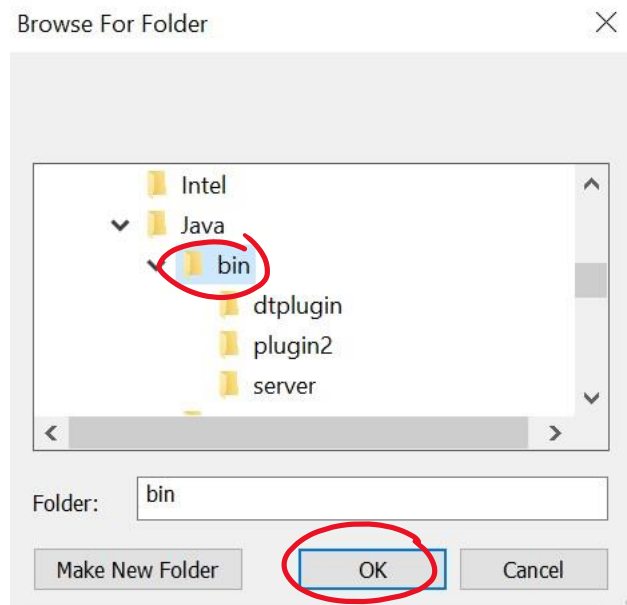
### 7.4.2. Pilih menu 'New' untuk membuat path Java



Gambar 1. 27 Menyunting Environment Variabel untuk Path Java

7.4.2.1. Pilih 'Browse' untuk mencari direktori Java di local disk C.

7.4.2.2. Klik 'bin' pada folder 'Java', setelah itu, 'Ok'.



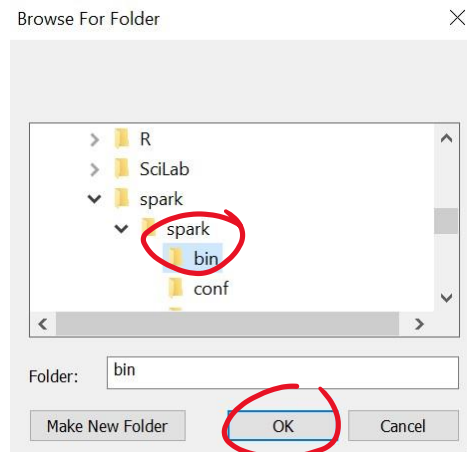
Gambar 1. 28 Seleksi bin Java

7.4.2.3. Lalu path untuk Java sudah terbentuk.

7.4.3. Pilih menu 'New' untuk membuat path Spark.

7.4.3.1. Klik browse untuk mencari folder 'spark' lalu pilih 'bin' Spark dan 'ok'.

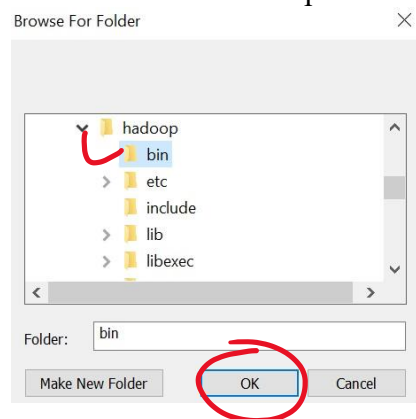




Gambar 1. 29 Seleksi bin Spark

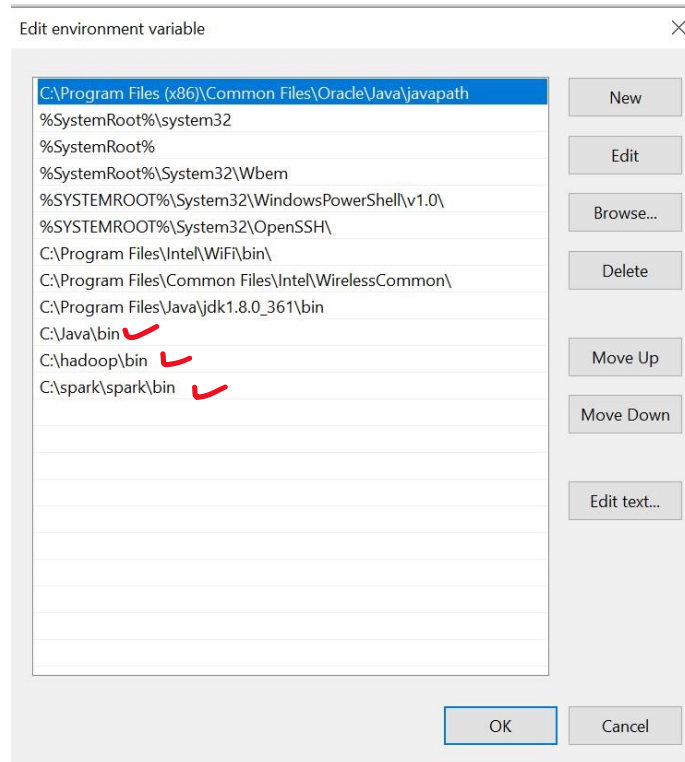
7.4.4. Pilih menu 'New' untuk membuat path Hadoop.

7.4.4.1. Klik browse untuk mencari 'bin' Hadoop lalu 'ok'.



Gambar 1. 30 Seleksi bin Hadoop

7.5. Nanti path Hadoop, Java, dan Path akan terbentuk seperti ini:



Gambar 1. 31 Hasil Membuat Path Java, Hadoop dan Spark

## 8. Memulai Hadoop dan Spark

### 8.1. Membuka Command prompter untuk menjalankan Hadoop & Spark

#### 8.1.1. Menjalankan Hadoop dengan menuliskan perintah 'hadoop'.

```

C:\Users\Lenovo>hadoop
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

Usage: hadoop [--config confdir] [--loglevel loglevel] COMMAND
where COMMAND is one of:
  fs                run a generic filesystem user client
  version           print the version
  jar <jar>         run a jar file
                   note: please use "yarn jar" to launch
                   YARN applications, not this command.
  checknative [-a|-h] check native hadoop and compression libraries availability
  conftest          validate configuration XML files
  distch path:owner:group:permission distributed metadata changer
  distcp <srcurl> <desturl> copy file or directories recursively
  archive -archiveName NAME -p <parent path> <src>* <dest> create a hadoop archive
  classpath         prints the class path needed to get the
                   Hadoop jar and the required libraries
  credential        interact with credential providers
  jnipath           prints the java.library.path
  kerbname          show auth_to_local principal conversion
  kdiag            diagnose kerberos problems
  key              manage keys via the KeyProvider
  trace            view and modify Hadoop tracing settings
  daemonlog        get/set the log level for each daemon
  or
  CLASSNAME        run the class named CLASSNAME

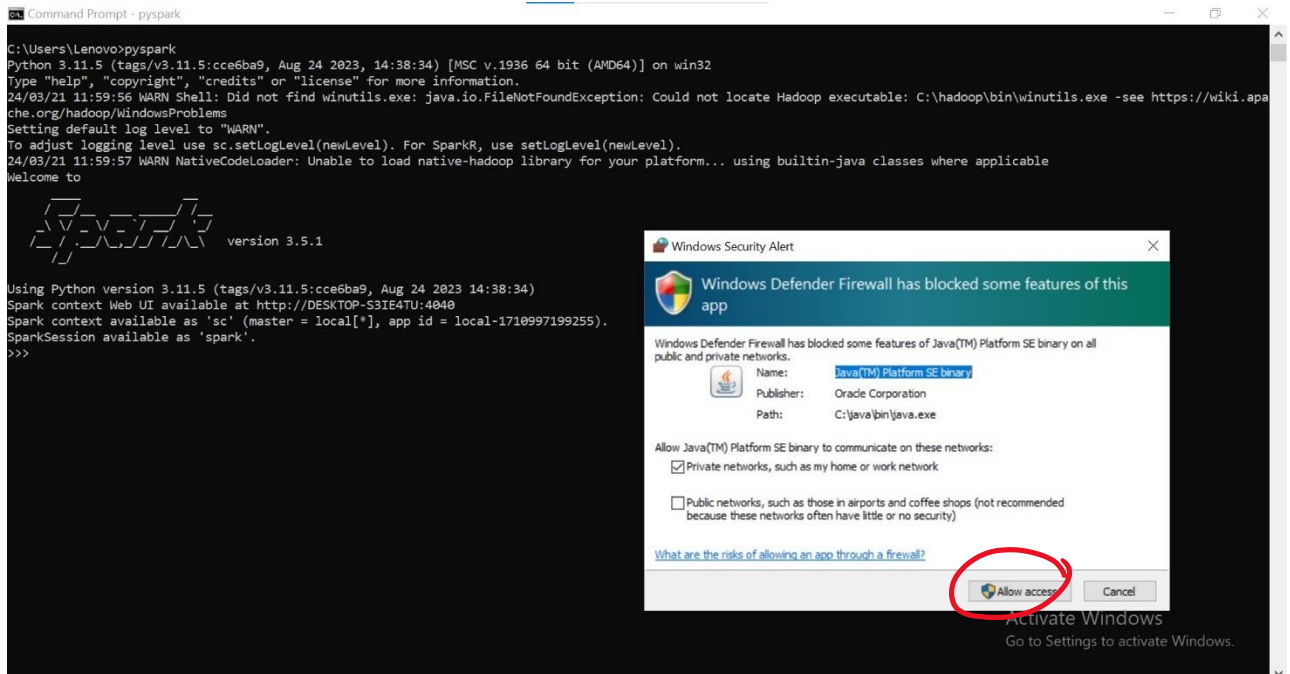
Most commands print help when invoked w/o parameters.

```

Gambar 1. 32 Menjalankan Hadoop di CMD

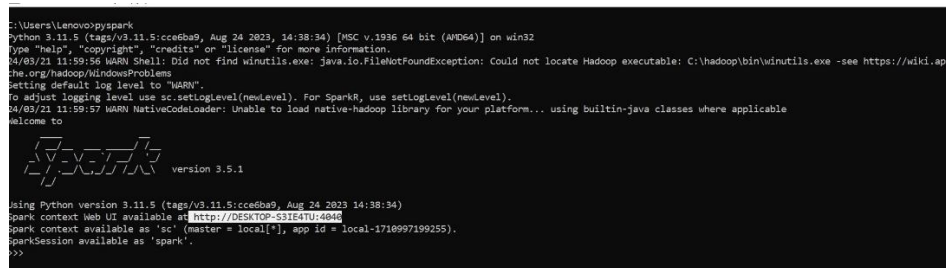
#### 8.1.2. Menjalankan Spark dengan menuliskan perintah 'pyspark'.

8.1.2.1. Setelah muncul versi dari Spark yaitu versi 3.5.1. akan muncul juga secara bersamaan Windows Security Alert. Klik "Allow access".



Gambar 1. 33 Menjalankan Spark di CMD

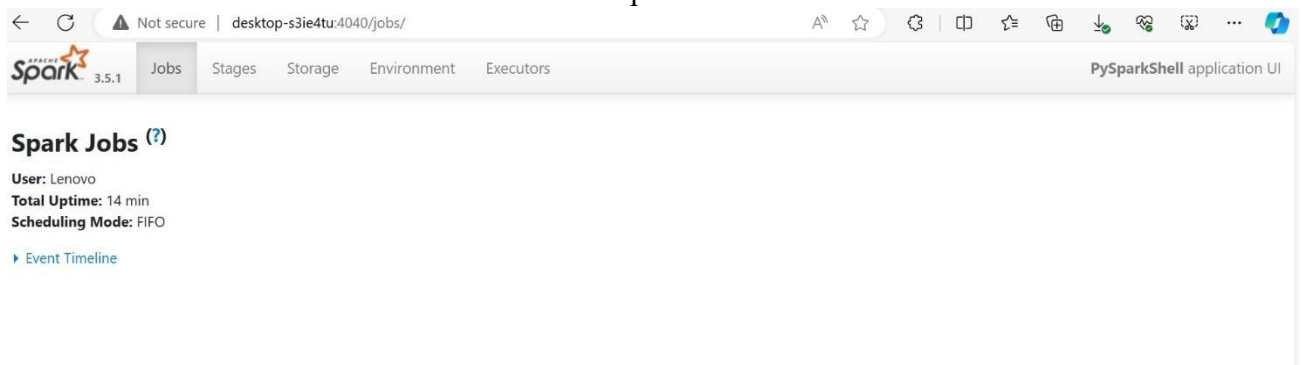
8.1.2.2. Pada cmd spark yang telah terbuka, copy tulisan 'http://DESKTOP-S3IE4TU:4040'.



Gambar 1. 34 Membuka Spark Melalui Surl Local Host pada CMD

8.1.2.3. Paste 'http://DESKTOP-S3IE4TU:4040' pada search engine.

8.1.2.4. Lalu akan muncul tampilan berikut.



Gambar 1. 35 Tampilan Spark di Website

8.1.2.5. Pilih menu , 'Executor'

8.1.2.6. Lalu akan muncul tampilan berikut.

## Executors

[Show Additional Metrics](#)

### Summary

	<div>▲</div> RDD Blocks <div>▼</div>	<div>▼</div> Storage Memory <div>▲</div>	<div>▼</div> Disk Used <div>▲</div>	<div>▼</div> Cores <div>▲</div>	<div>▼</div> Active Tasks <div>▲</div>	<div>▼</div> Failed Tasks <div>▲</div>	<div>▼</div> Complete Tasks <div>▲</div>	<div>▼</div> Total Tasks <div>▲</div>	<div>▼</div> Task Time (GC Time) <div>▲</div>	<div>▼</div> Input <div>▲</div>	<div>▼</div> Shuffle Read <div>▲</div>	<div>▼</div> Shuffle Write <div>▲</div>	<div>▼</div> Excluded <div>▲</div>
Active(1)	0	0.0 B / 366.3 MiB	0.0 B	8	0	0	0	0	16 min (95.0 ms)	0.0 B	0.0 B	0.0 B	0
Dead(0)	0	0.0 B / 0.0 B	0.0 B	0	0	0	0	0	0.0 ms (0.0 ms)	0.0 B	0.0 B	0.0 B	0
Total(1)	0	0.0 B / 366.3 MiB	0.0 B	8	0	0	0	0	16 min (95.0 ms)	0.0 B	0.0 B	0.0 B	0

### Executors

Show 20 

▼

 entries

Search:

Gambar 1. 36 Tampilan Menu Executors Spark di Website

8.1.2.7. Selesai.

## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1. Hasil Praktikum

Dalam eksperimen ini, instalasi Apache Hadoop di Windows 10 telah berhasil saya lakukan dengan mengikuti langkah-langkah yang tercantum dalam tutorial resmi Hadoop. Walaupun proses instalasi secara umum berjalan tanpa kendala, beberapa tantangan muncul terutama dalam mengatur konfigurasi awal dan menangani masalah yang timbul untuk memastikan komponen Hadoop dapat berfungsi secara optimal. Salah satu hal yang cukup menantang adalah penyesuaian variabel lingkungan dan konfigurasi file Hadoop agar sesuai dengan persyaratan Windows 10. Namun, dengan bantuan referensi tambahan dan kerjasama dengan partner saya, akhirnya saya berhasil menyelesaikan instalasi dengan sukses.

#### 4.2. Pembahasan

Melalui eksperimen ini, saya mendapat pemahaman yang lebih mendalam tentang konsep dasar infrastruktur Big Data dan langkah-langkah instalasi Hadoop. Saya juga menyadari pentingnya pemecahan masalah dan kolaborasi dengan orang lain dalam mengatasi kendala teknis.

Lalu ada beberapa hal yang saya sadari mengenai Hadoop ini. Beberapa hal tersebut di antaranya adalah sebagai berikut:

1. Apa peran Java dan Spark saat kita menggunakan Hadoop?

Java adalah bahasa pemrograman yang umum digunakan dalam ekosistem Hadoop karena Hadoop sendiri ditulis menggunakan Java. Oleh karena itu, Java adalah salah satu bahasa yang paling kompatibel dengan Hadoop dan berfungsi dengan baik dalam mengembangkan aplikasi dan skrip yang berinteraksi dengan Hadoop.

Spark, di sisi lain, adalah platform pemrosesan data cepat yang juga terintegrasi dengan Hadoop. Spark menyediakan API yang mudah digunakan untuk melakukan pemrosesan data paralel di atas Hadoop, yang memungkinkan kinerja yang lebih cepat dan lebih efisien daripada MapReduce yang digunakan secara tradisional dalam Hadoop.

2. Mengapa kita mengatur Spark dan Java sebelum menggunakan Hadoop?

Sebelum menggunakan Hadoop, kita harus mengatur lingkungan kerja untuk Spark dan Java karena Hadoop sering kali memerlukan infrastruktur yang sesuai dan terhubung dengan perangkat lunak dan bahasa pemrograman tertentu untuk berinteraksi dengan sistem secara efektif. Spark dan Java digunakan untuk membangun aplikasi dan skrip yang akan berjalan di atas Hadoop untuk melakukan pemrosesan data.

3. Apa kegunaan membuat variabel user dan value variabel untuk Hadoop, Java & Spark?

Membuat variabel pengguna (user) dan nilai variabel (value) adalah praktik umum dalam konfigurasi lingkungan kerja. Ini memungkinkan kita untuk dengan mudah mengakses dan menggunakan nilai-nilai ini di seluruh kode kita tanpa harus secara harfiah menuliskan nilai-nilai itu berulang-ulang. Misalnya, kita dapat menggunakan variabel pengguna (user) untuk menyimpan nama pengguna yang digunakan untuk mengakses Hadoop, Java, atau Spark, sementara nilai variabel (value) dapat berisi informasi seperti path ke instalasi perangkat lunak tersebut.

4. Mengapa kita membuat path dulu sebelum menggunakan Hadoop, Java, dan Spark?

Membuat path (jalur) adalah langkah penting dalam konfigurasi lingkungan kerja karena memastikan bahwa sistem operasi dapat menemukan dan menjalankan perangkat lunak yang diperlukan. Dalam konteks Hadoop, Java, dan Spark, membuat path memungkinkan sistem operasi untuk menemukan file biner, pustaka, dan alat yang diperlukan untuk menjalankan aplikasi dan skrip yang berkaitan dengan teknologi-teknologi ini. Tanpa path yang benar, sistem operasi mungkin tidak akan dapat menemukan perangkat lunak yang diperlukan dan kita akan menghadapi kesalahan saat mencoba menjalankan aplikasi.

Membuat path sebelum menggunakan Hadoop, Java, dan Spark adalah penting karena ini memungkinkan sistem operasi kita untuk menemukan lokasi di mana file biner dan perpustakaan terkait dengan perangkat lunak tersebut disimpan. Ini memberikan akses mudah, eksekusi perintah yang lancar, kemudahan dalam pengembangan aplikasi, integrasi yang baik dengan lingkungan, dan menghindari konflik versi. Dengan mengatur PATH, kita dapat dengan mudah mengakses perintah-perintah dan alat terkait seperti hadoop, java, atau spark-submit langsung dari baris perintah tanpa harus menentukan jalur lengkap ke file biner. Ini juga mempermudah penggunaan berbagai perpustakaan dan alat yang diperlukan dalam proses pengembangan dan pengujian aplikasi menggunakan Hadoop, Java, atau Spark. Selain itu, dengan mengatur PATH dengan benar, kita memastikan bahwa aplikasi atau framework ini dapat menemukan dependensi yang diperlukan saat dijalankan, serta menghindari potensi konflik versi dengan menggunakan versi yang tepat dari Java, Hadoop, dan Spark. Membuat path juga memastikan bahwa aplikasi atau framework dapat dijalankan dari mana saja di sistem kita tanpa perlu mengganti setelan setiap kali kita ingin mengaksesnya, meningkatkan fleksibilitas dan kenyamanan dalam penggunaan Java, Hadoop, dan Spark di lingkungan kita.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Setelah melakukan praktikum instalasi Apache Hadoop di Windows 10, saya dapat menyimpulkan bahwa proses instalasi Hadoop memerlukan pemahaman yang baik tentang konsep dasar Big Data serta keterampilan dalam menangani masalah teknis yang mungkin muncul. Meskipun terdapat beberapa tantangan dalam mengatur konfigurasi awal dan menangani masalah teknis, kerjasama dengan partner serta referensi tambahan dapat membantu mengatasi kendala tersebut.

Selain itu, saya menyadari pentingnya mempersiapkan diri dengan baik sebelum melakukan instalasi dan mengikuti petunjuk dengan cermat. Instalasi Hadoop di lingkungan Windows 10 memerlukan penyesuaian tambahan untuk memastikan kinerja yang optimal. Oleh karena itu, saya mendorong untuk terus menjelajahi opsi konfigurasi yang tersedia dan memperdalam pemahaman tentang penggunaan Hadoop dalam lingkungan Windows 10. Dengan demikian, praktikum ini memberikan pengalaman yang berharga dalam memahami konsep Big Data serta penerapannya melalui instalasi Hadoop.

#### 5.2. Saran

Rekomendasi saya bagi mereka yang tertarik untuk menginstal Hadoop 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

Khan, N., Yaqoob, I., Hashem, I., Inayat, Z., Ali, W., Alam, M., Shiraz, M., & Gani, A. (2014). Big Data: Survey, Technologies, Opportunities, and Challenges. The Scientific World Journal, 2014. <https://doi.org/10.1155/2014/712826>.

Dev, D., & Patgiri, R. (2016). A Survey of Different Technologies and Recent Challenges of Big Data. , 537-548. [https://doi.org/10.1007/978-81-322-2529-4\\_56](https://doi.org/10.1007/978-81-322-2529-4_56).

Mall, N., , S., & Rana, S. (2016). Overview of Big Data and Hadoop. Imperial journal of interdisciplinary research, 2.

Bhandarkar, M. (2010). MapReduce programming with apache Hadoop. 2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS), 1-1. <https://doi.org/10.1109/IPDPS.2010.5470377>.

RevoU. (2024). Big Data. Diakses pada 18 Maret 2024, dari <https://revou.co/kosakata/big-data>

Sari, R. P. (2024, 30 Januari). Mengenal Apa Itu Hadoop? Solusi dalam Era Big Data Analytics. Diakses pada 18 Maret 2024, dari <https://www.cloudcomputing.id/pengetahuan-dasar/mengenal-hadoop-bigdata>

White, T. (2012). Hadoop: The Definitive Guide. O'Reilly Media.