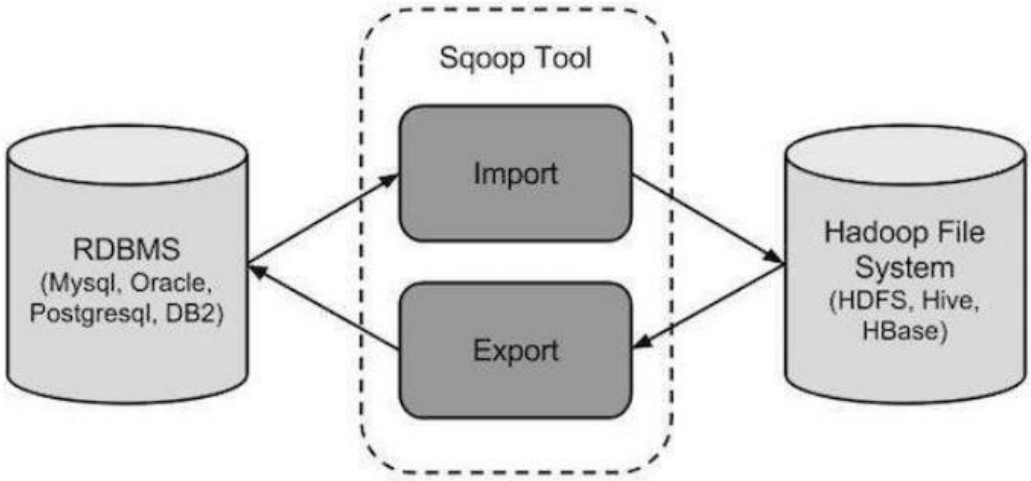
MODUL 5 – IMPORT DAN EXPORT DATA DENGAN SQOOP

1.1. Deskripsi Singkat

Apache Sqoop merupakan tool yang memiliki dua fungsi utama yaitu impor dan ekspor data terutama dalam penggunaan HDFS untuk penyimpanan data. Secara umum alur impor dan ekspor data dengan menggunakan Sqoop adalah sebagai berikut.



Impor data dilakukan dari RDBMS ke HDFS dimana setiap baris pada tabel akan menjadi record pada HDFS. Semua record disimpan sebagai data dalam fomat text files atau sebagai binary (seperti Avro dan sequence files). Sebaliknya, ekspor data dilakukan dari HDFS ke RDBMS dengan melakukan parsing pada input data dari HDFS dan menyesuaikan delimiter yang digunakan sesuai dengan kebutuhan.

1.2. Tujuan Praktikum

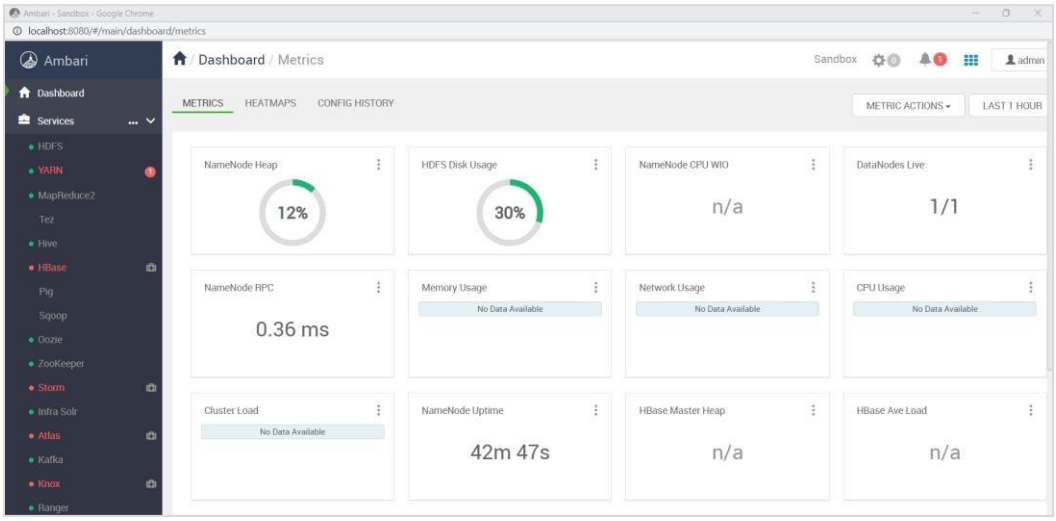
Setelah praktikum pada modul 5 ini, diharapkan mahasiswa mempunyai kompetensi dalam melakukan impor dan ekspor data sesuai dengan menggunakan sqoop.

1.3. Material Praktikum

Persyaratan yang dibutuhkan untuk melakukan praktikum 5 yaitu :

1. HDP yang telah terinstal pada VirtualBox.

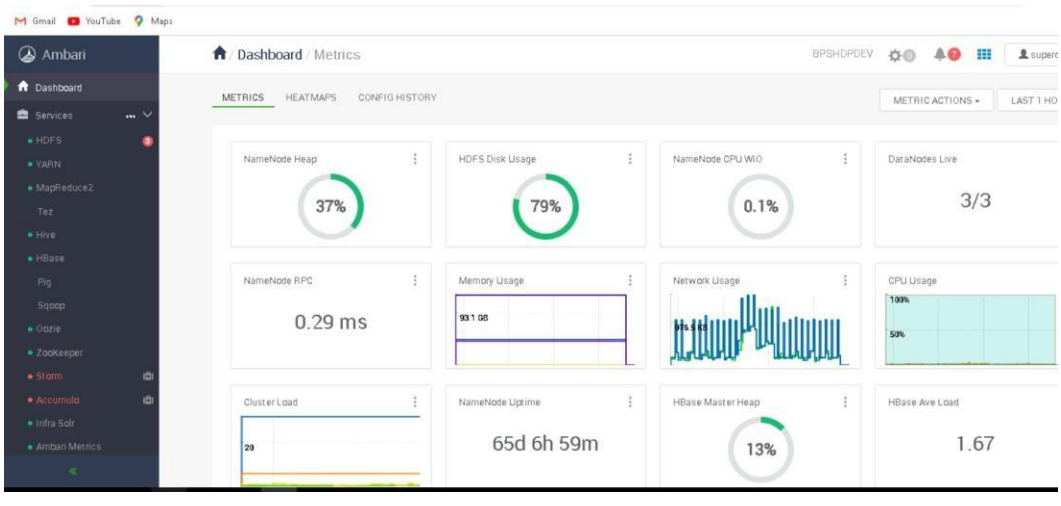
2. Ambari dapat diakses



3. Sercice Sqoop dalam keadaan hidup (*on*) dan dapat digunakan.
4. RDBMS yang telah siap digunakan (contoh yang digunakan pada modul praktikum ini adalah SQL Server). **Jika SQL Server belum tersedia di Lab komputer, maka dapat menggunakan RDBMS yang ada, namun koneksi yang digunakan pada sqoop harus disesuaikan dengan RDBMS yang digunakan**.

1.4. Kegiatan Praktikum

1. Akses Ambari dengan menggunakan akun yang telah dibuat sebelumnya untuk memastikan service yang dibutuhkan dalam keadaan hidup dan siap digunakan, terutama HDFS, Yarn, MapReduce2, Sqoop, Oozie, Zookeeper.



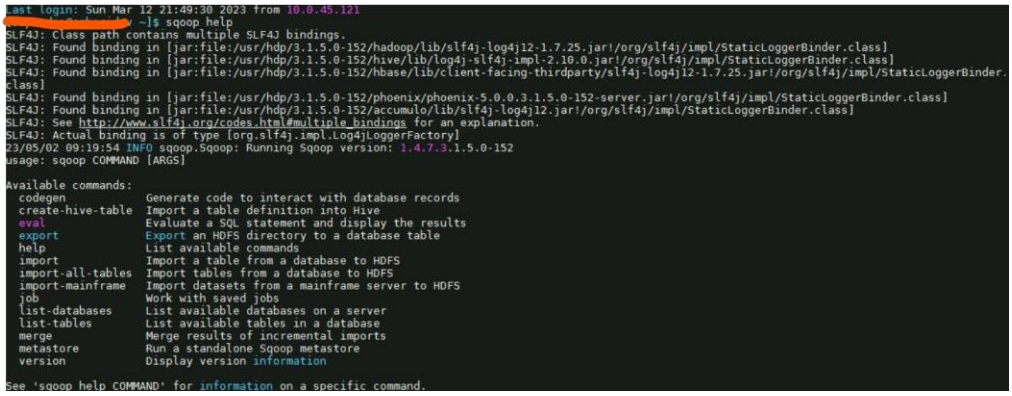
2. Untuk memastikan Sqoop telah terinstal dan siap digunakan, lakukan langkah-langkah berikut.
    1) Akses http://localhost:4200 pada browser.
    2) Lakukan login terlebih dahulu.

Username: **root**

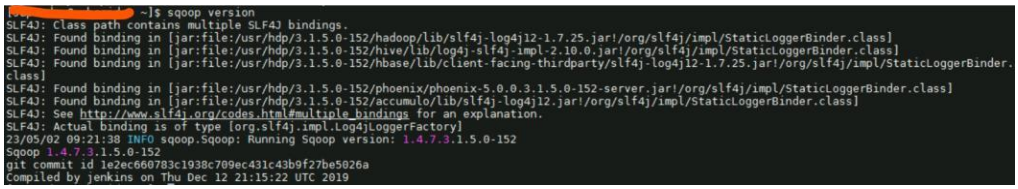
Existing Password: **Tp4stis**

3) Jalankan command:

a. **`sqoop help`** untuk mengetahui command apa saja yang dapat digunakan



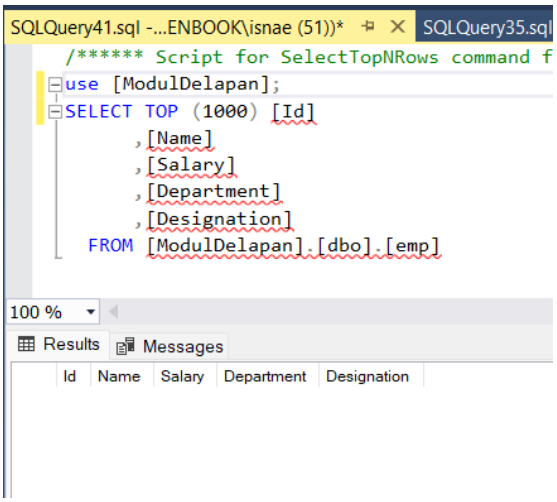
b. **`sqoop version`** untuk mengetahui versi dari sqoop yang terinstal pada HDP sandbox



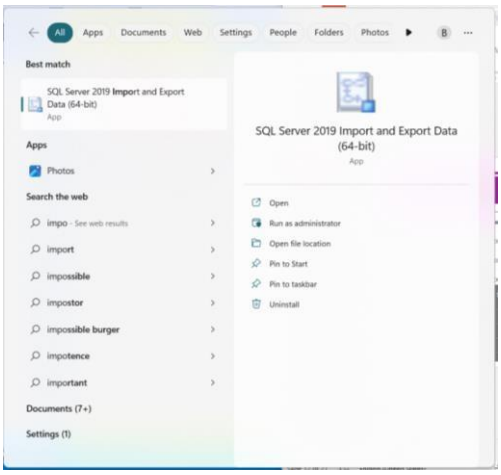
3. Buat database di SQL Server atau MySql atau repositori yang tersedia di PC laboratorium komputer dengan nama ModulDelapan.
4. Buat tabel dengan struktur sebagai berikut:

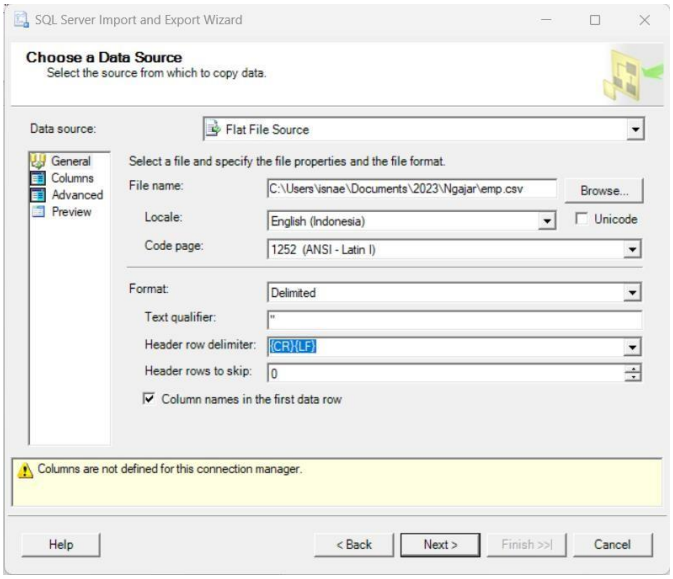
```
Create table emp
(
        [Id] [int] Identity(1,1) not null,
        [Name] [varchar](255) not null,
        [Salary] [int] not  null,
        [Department] [varchar](255) not null,
        [Designation] [varchar](255) not null
)
```

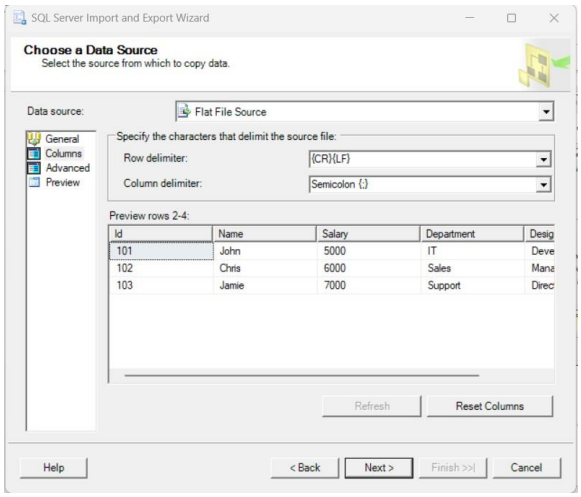
Jalankan DDL di atas, sehingga terbentuk tabel **`emp`** pada database (sesuaikan SQL dengan RDBMS).

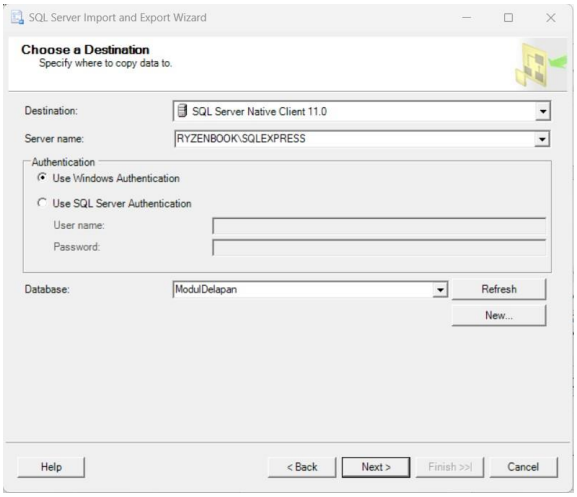


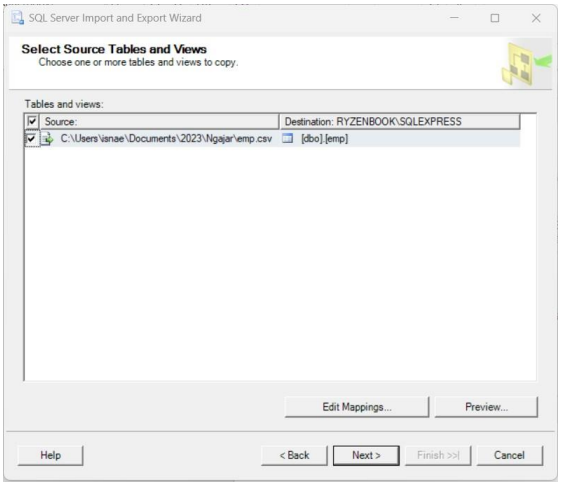
5. Lakukan pengisian data pada tabel yang telah disiapkan. Untuk impor data, dapat dilakukan dengan menggunakan fitur Import and Export pada SQL server atau menggunakan SSIS (sebagaimana yang telah dipelajari pada modul 5).

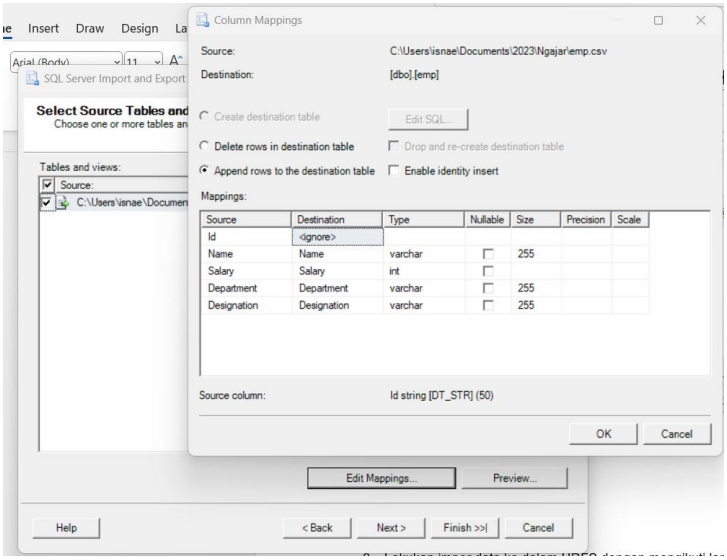


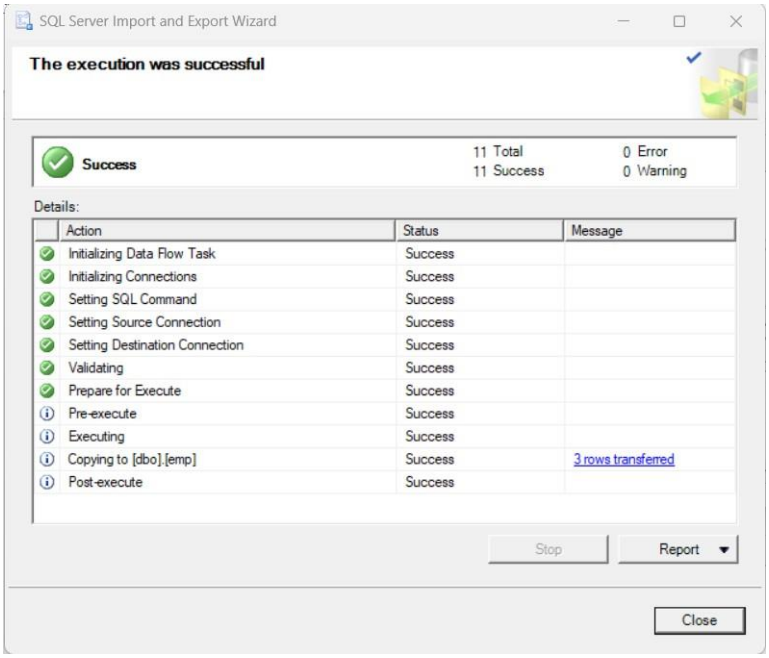



SQL Server Import and Export Wizard
Choose a Data Source
Select the source from which to copy data.
Data source:
Flat File Source
General
Columns
Advanced
Preview
Specify the characters that delimit the source file:
Row delimiter:
{CR}{LF}
Column delimiter:
Semicolon (;)
Preview rows 2-4:
Id
Name
Salary
Department
Desig
101
John
5000
IT
Deve
102
Chris
6000
Sales
Mana
103
Jamie
7000
Support
Direc
Refresh
Reset Columns
Help
< Back
Next >
Finish >>|
Cancel


SQL Server Import and Export Wizard
Choose a Destination
Specify where to copy data to.
Destination:
SQL Server Native Client 11.0
Server name:
RYZENBOOK\SQLEXPRESS
Authentication
Use Windows Authentication
Use SQL Server Authentication
User name:
Password:
Database:
ModulDelapan
Refresh
New...
Help
< Back
Next >
Finish >>|
Cancel


SQL Server Import and Export Wizard
Select Source Tables and Views
Choose one or more tables and views to copy.
Tables and views:
Source:
Destination: RYZENBOOK\SQLEXPRESS
C:\Users\snae\Documents\2023\Ngajar\emp.csv
[dbo].[emp]
Edit Mappings...
Preview...
Help
< Back
Next >
Finish >>|
Cancel





	Id	Name	Salary	Department	Designation
1	1	John	5000	IT	Developer
2	2	Chris	6000	Sales	Manager
3	3	Jamie	7000	Support	Director

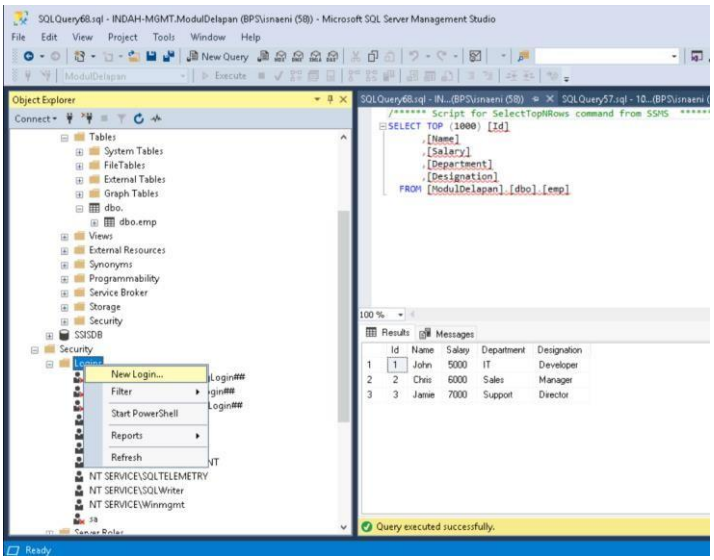
Untuk DBMS MySQL gunakan command berikut untuk membuat tabel dan mengisi tabel:

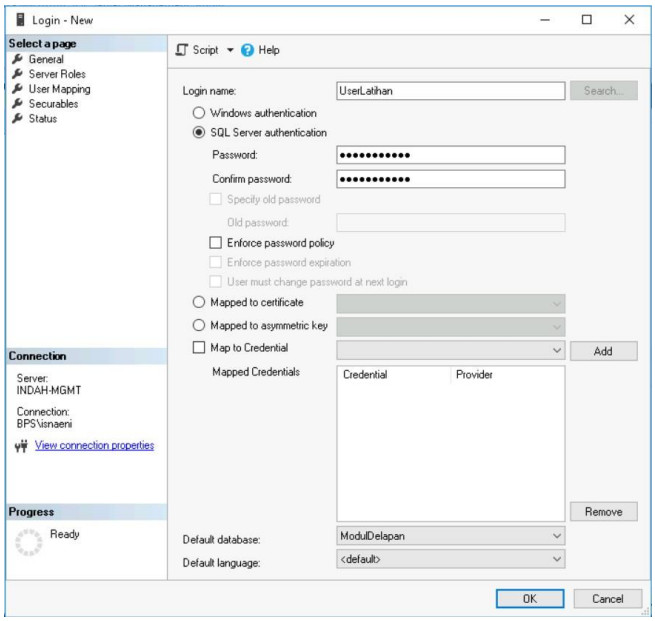
```
Create table emp
(
      Id int not null AUTO_INCREMENT,
```

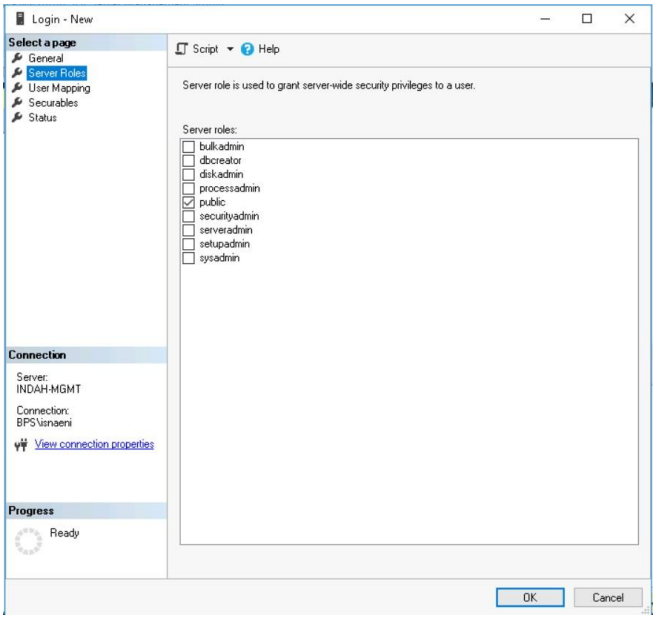
```
      Name varchar(255) not null,
      Salary int not null,
      Department varchar(255) not null,
      Designation varchar(255) not null,
    PRIMARY KEY (Id)
);

INSERT     INTO     emp(Id,Name,Salary,Department,Designation)
VALUES(1,'John','5000','IT','Developer');
INSERT     INTO     emp(Id,Name,Salary,Department,Designation)
VALUES(2,'Chris','6000','Sales','Manager');
INSERT     INTO     emp(Id,Name,Salary,Department,Designation)
VALUES(3,'Jamie','7000','Support','Director');
```

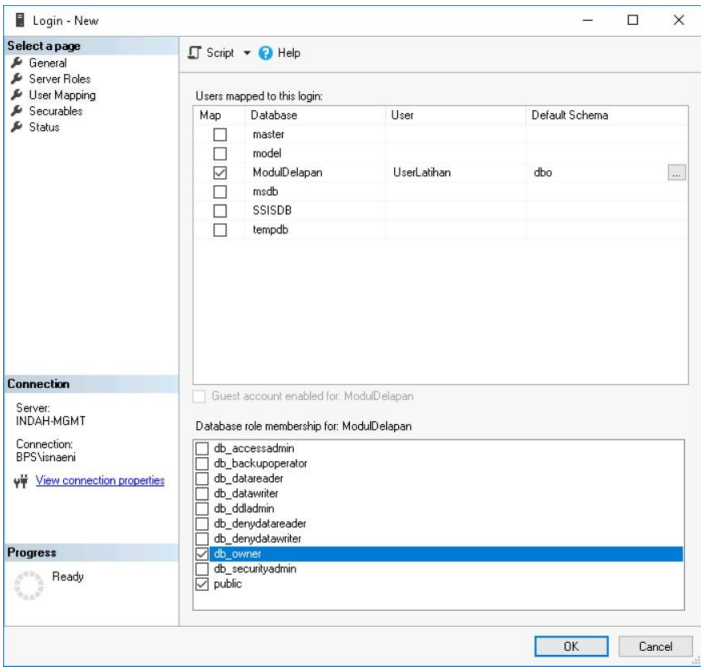
6. Lakukan impor data ke dalam HDFS dengan mengikuti langkah-langkah berikut.
   1) Impor data dari satu tabel

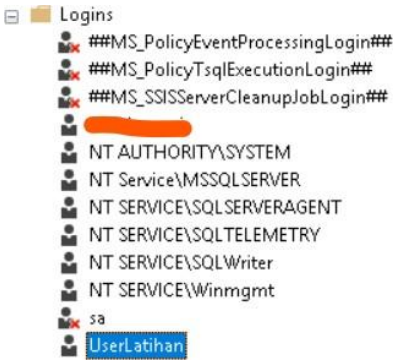
      Untuk melakukan proses ini, akun database yang digunakan adalah user dengan sql authentication. Untuk membuat sql authentication pada sql server, ikuti langkah berikut.




Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Login name:
UserLatihan
Search...
Windows authentication
SQL Server authentication
Password:
Confirm password:
Specify old password
Old password:
Enforce password policy
Enforce password expiration
User must change password at next login
Mapped to certificate
Mapped to asymmetric key
Map to Credential
Add
Mapped Credentials
Credential
Provider
Remove
Default database:
ModulDelapan
Default language:
<default>
Connection
Server:
INDAH-MGMT
Connection:
BPS\isnaeni
View connection properties
Progress
Ready
OK
Cancel


Login - New
Select a page
General
Server Roles
User Mapping
Securables
Status
Script
Help
Server role is used to grant server-wide security privileges to a user.
Server roles:
bulkadmin
dbcreator
diskadmin
processadmin
public
securityadmin
serveradmin
setupadmin
sysadmin
Connection
Server:
INDAH-MGMT
Connection:
BPS\isnaeni
View connection properties
Progress
Ready
OK
Cancel

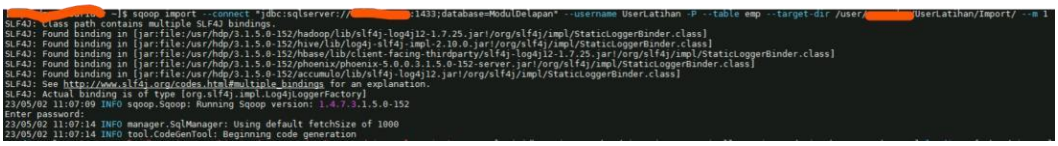





```
[root@sandbox-hdp ~]# hdfs dfs -mkdir sdata/
[root@sandbox-hdp ~]# hdfs dfs -ls
Found 7 items
drwx------   - root hdfs          0 2023-02-15 01:54 .Trash
drwx------   - root hdfs          0 2023-03-19 03:40 .staging
drwxr-xr-x   - root hdfs          0 2023-02-14 18:44 geo_data
drwxr-xr-x   - root hdfs          0 2023-03-19 03:24 input
drwxr-xr-x   - root hdfs          0 2023-03-19 03:40 output
drwxr-xr-x   - root hdfs          0 2023-03-19 04:06 sdata
drwxr-xr-x   - root hdfs          0 2023-02-14 20:13 tdata
```

Jalankan command berikut (SQL Server):

```
sqoop import --connect "jdbc:sqlserver://[ip/nama server]:1433; database:ModulDelapan" --username UserLatihan -P --table emp --target-dir /user/[path]/ --m 1
```



Command import data untuk MySQL:

```
sqoop import --connect "jdbc:mysql://10.0.2.2:3306/[nama db]" --username [user] -password [password] --table [nama table] --target-dir [path] --m 1
```

Hasil import tabel **emp** dapat dilihat pada fitur Files View pada Ambari:

2) Incremental load

Henning, 4000, IT, Developer

Tambahkan record di atas pada tabel **emp**.

	Id	Name	Salary	Department	Designation
1	1	John	5000	IT	Developer
2	2	Chris	6000	Sales	Manager
3	3	Jamie	7000	Support	Director
4	4	Henning	4000	IT	Developer

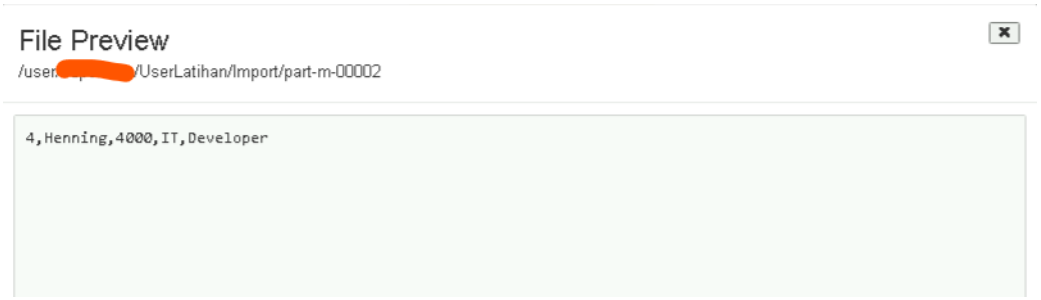
Jalankan command berikut untuk SQL Server:

```
sqoop import --connect "jdbc:sqlserver://[ip/nama server]:1433; database:ModulDelapan" --username UserLatihan -P --table emp --target-dir /user/[path]/ --m 1--incremental append --check-column id --last-value 3
```

Command import data incremental untuk MySQL:

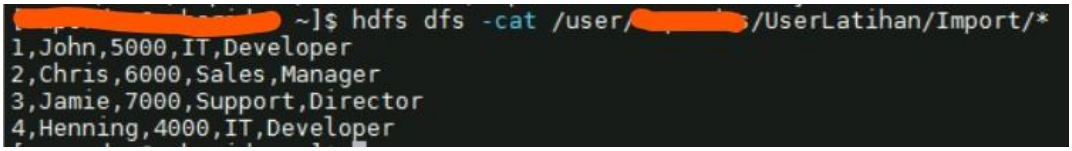
```
sqoop import --connect "jdbc:mysql://10.0.2.2:3306/[nama db]" --username [user] -password [password] --table [nama table] --target-dir [path] --m 1 --incremental append --check-column id --last-value 3
```

Hasil import tabel **emp** secara incremental dapat dilihat pada fitur Files View pada Ambari:



3) Read data hasil import secara keseluruhan

```
hdfs dfs -cat [path data]
```



7. Lakukan ekspor data ke dalam HDFS dengan mengikuti langkah-langkah berikut.

1) Create table

```
USE [ModulDelapan]
GO

/****** Object:  Table [dbo].[emp2]    Script Date: 02/05/2023 14.01.43 ******/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[emp2](
    [Id] [varchar](255) NOT NULL,
    [Name] [varchar](255) NOT NULL,
    [Salary] [varchar](255) NOT NULL,
    [Department] [varchar](255) NOT NULL,
    [Designation] [varchar](255) NOT NULL
) ON [PRIMARY]
GO
```

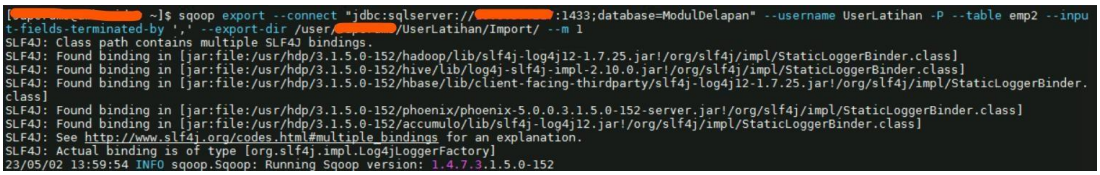
2) Jalan command berikut untuk SQL Server:

```
sqoop export --connect "jdbc:sqlserver://[ip/nama
server]:1433; database:ModulDelapan" --username
```

```
UserLatihan -P --table emp2 --export-dir /user/[path]/ --
m 1
```

Command export data untuk MySQL:

```
sqoop export --connect "jdbc:mysql://10.0.2.2:3306/[nama
db]" --username [username] -password [password] --table
[nama table] --export-dir [path]/* --m 1
```

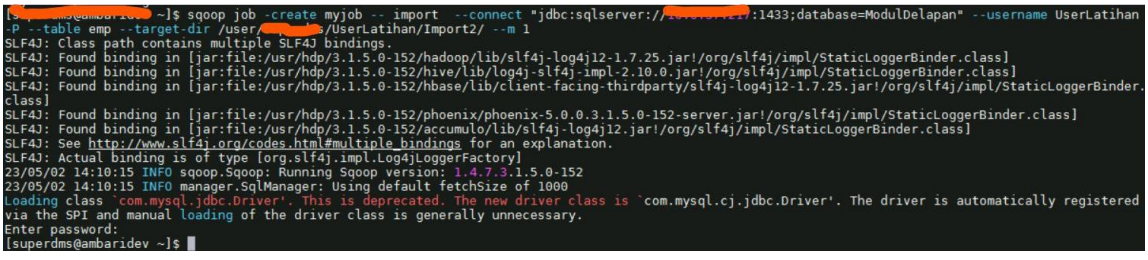


Cek hasil ekspor:

Id	Name	Salary	Department	Designation
1	John	5000	IT	Developer
2	Chris	6000	Sales	Manager
3	Jamie	7000	Support	Director
4	Henning	4000	IT	Developer

8. Membuat sqoop job

```
sqoop job -create [nama job] --import --connect
“jdbc:sqlserver://[ip/nama server]:1433;
database:ModulDelapan” --username UserLatihan -P --table emp
--target-dir /user/[path]/ --m 1
```



Eksekusi sqoop job:

```
sqoop job –exec [nama job]
```

```
[redacted] ~]$ sqoop job --exec myjob
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/phoenix/phoenix-5.0.0.3.1.5.0-152-server.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/accumulo/lib/slf4j-log4j12.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
23/05/02 14:31:12 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7.3.1.5.0-152
```

Hasil eksekusi job:

```
[redacted] ~]$ hdfs dfs -cat /user/superdms/UserLatihan/Import2/*
1,John,5000,IT,Developer
2,Chris,6000,Sales,Manager
3,Jamie,7000,Support,Director
4,Henning,4000,IT,Developer
```

9. Menggunakan sqoop command lainnya

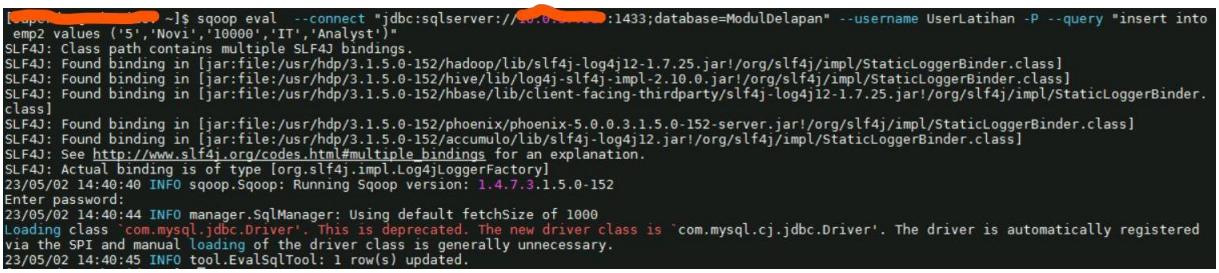
```
sqoop list-tables --connect "jdbc:sqlserver://[ip/nama server]:1433; database:ModulDelapan" --username UserLatihan -P
```

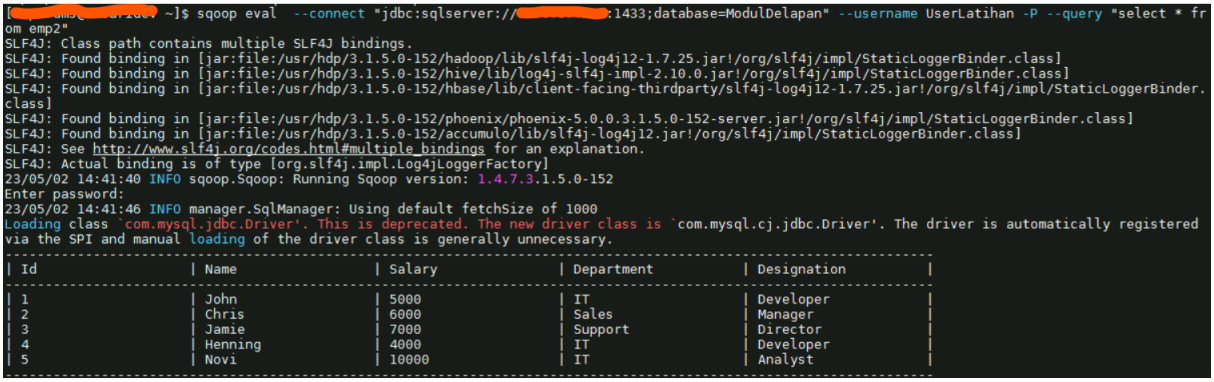
```
[redacted] ~]$ sqoop list-tables  --connect "jdbc:sqlserver://[redacted]:1433;database=ModulDelapan" --username UserLatihan -P
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder
class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/phoenix/phoenix-5.0.0.3.1.5.0-152-server.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/accumulo/lib/slf4j-log4j12.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
23/05/02 14:35:43 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7.3.1.5.0-152
Enter password:
23/05/02 14:35:47 INFO manager.SqlManager: Using default fetchSize of 1000
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered
via the SPI and manual loading of the driver class is generally unnecessary.
emp
emp2
```

```
sqoop eval --connect "jdbc:sqlserver://[ip/nama server]:1433; database:ModulDelapan" --username UserLatihan -P -query "select * from emp2"
```

```
[redacted] ~]$ sqoop eval  --connect "jdbc:sqlserver://[redacted]:1433;database=ModulDelapan" --username UserLatihan -P --query "select * fr
om emp2"
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/hbase/lib/client-facing-thirdparty/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/phoenix/phoenix-5.0.0.3.1.5.0-152-server.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.1.5.0-152/accumulo/lib/slf4j-log4j12.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
23/05/02 14:38:13 INFO sqoop.Sqoop: Running Sqoop version: 1.4.7.3.1.5.0-152
Enter password:
23/05/02 14:38:19 INFO manager.SqlManager: Using default fetchSize of 1000
Loading class `com.mysql.jdbc.Driver'. This is deprecated. The new driver class is `com.mysql.cj.jdbc.Driver'. The driver is automatically registered
via the SPI and manual loading of the driver class is generally unnecessary.
-----------------------------------------------------------------------------------------------------------
| Id                   | Name                 | Salary               | Department           | Designation          |
-----------------------------------------------------------------------------------------------------------
| 1                    | John                 | 5000                 | IT                   | Developer            |
| 2                    | Chris                | 6000                 | Sales                | Manager              |
| 3                    | Jamie                | 7000                 | Support              | Director             |
| 4                    | Henning              | 4000                 | IT                   | Developer            |
-----------------------------------------------------------------------------------------------------------
```

```
sqoop eval --connect "jdbc:sqlserver://[ip/nama server]:1433;
database:ModulDelapan" --username UserLatihan -P -query "insert
into emp2 values ('5','Novi','10000','IT','Analyst')"
```





1.5. Penugasan

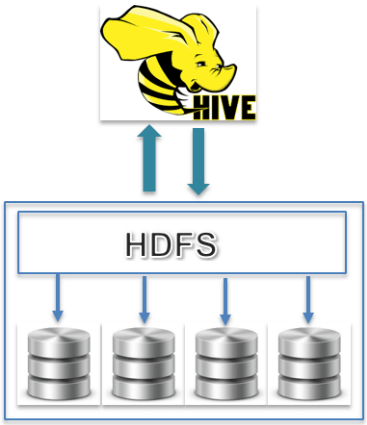
Kerjakan sesuai dengan yang dijelaskan pada bagian Kegiatan Praktikum untuk data yang berbeda yaitu Employee Sample Data.csv. Hasil pekerjaan praktikum berupa dokumentasi:

1. Proses dan hasil impor data ke HDFS.
2. Proses incremental load untuk data berikut sekaligus hasil *incremental load*-nya (untuk record yang akan di-*load* dapat dibuat sendiri)
3. Proses ekspor data:
   - Create tabel
   - Proses ekspor data dari HDFS ke RDBMS
   - Hasil ekspor data
4. Pembuatan job untuk import data ke hdfs dan hasil eksekusinya.
5. Proses insert data dengan menggunakan `sqoop eval`.

MODUL 9 – HIVE

Deskripsi Singkat

Apache Hive adalah sistem data warehouse terdistribusi dan toleran terhadap kesalahan yang memungkinkan analitik dalam skala besar dan memfasilitasi pembacaan, penulisan, dan pengelolaan petabyte data yang berada di penyimpanan terdistribusi menggunakan SQL. Secara umum proses query menggunakan Hive adalah sebagai berikut.



Apache Hive merupakan komponen dari Hortonworks Data Platform (HDP). Hive menyediakan antar muka dalam bentuk query seperti SQL dimana data disimpan dalam HDP dengan sistem file HDFS. Data yang dapat di inputkan dalam Hadoop dapat disimpan dalam berbagai format data seperti format seperti comma separated value, parquet, Avro ataupun format data lainnya.

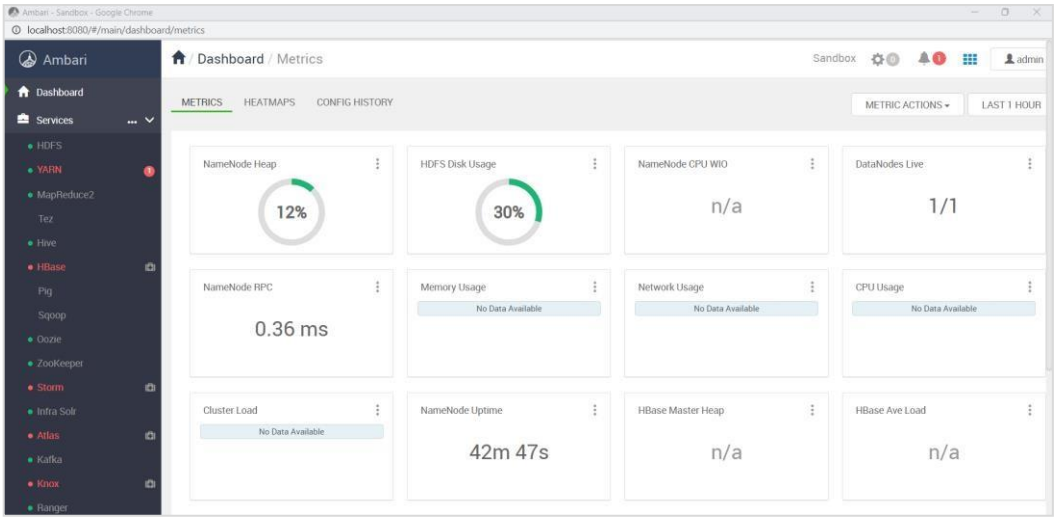
Tujuan Praktikum

Setelah praktikum pada modul 9 ini, diharapkan mahasiswa mempunyai kompetensi dalam melakukan query atau analisis data dengan menggunakan Hive.

Material Praktikum

Persyaratan yang dibutuhkan untuk melakukan praktikum 9 yaitu:

1. HDP yang telah terinstal pada VirtualBox.
2. Ambari dapat diakses



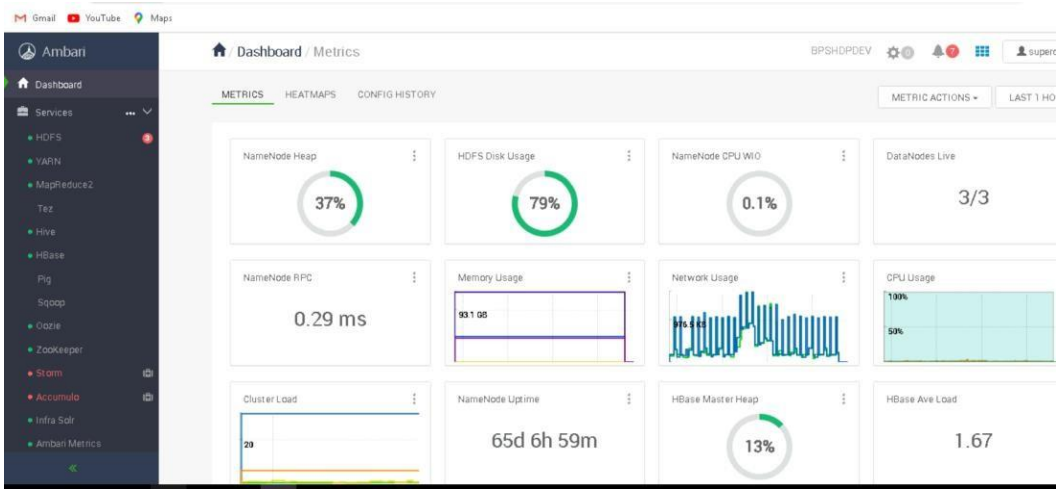
3. Service Hive dalam keadaan hidup (*on*) dan dapat digunakan.
4. Service Data Analytics Studio (DAS) dalam keadaan hidup (*on*) dan dapatdigunakan.

[redacted]iatan Praktikum

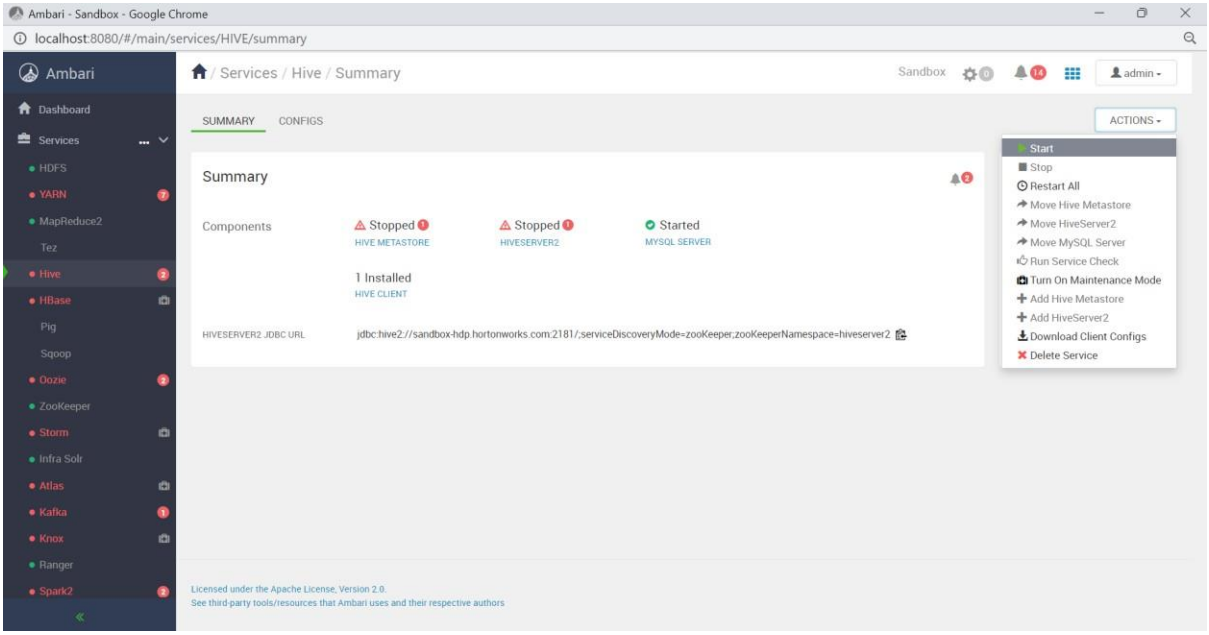
A. Persiapan

Untuk memastikan Hive nantinya dapat dijalankan, lakukan langkah-langkah sebagai berikut:

1. Akses Ambari dengan menggunakan akun yang telah dibuat sebelumnya untuk memastikan service yang dibutuhkan dalam keadaan hidup dan siap digunakan, terutama Hive dan Data Analytics Studio (DAS), HDFS, Yarn, MapReduce2, Sqoop, Oozie, Zookeeper.



Jika belum aktif, dapat klik tombol Action pada kanan atas layer. Kemudian klik Start. Lakukan untuk beberapa service yang diperlukan.



2. Untuk memastikan Hive telah terinstal dan siap digunakan, lakukan langkah-langkah berikut.
    1) Akses http://localhost:4200 pada browser.
    2) Lakukan login terlebih dahulu.
    Username: **root**
    Existing Password: **Tp4stis**
    3) Jalankan command:
        a. **`hive -H`** atau **`hive --help`** untuk mengetahui command apa saja yang dapat digunakan

```
[hdfs@sandbox-hdp ~]$ hive -H
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Unrecognized option: -H
Usage: java org.apache.hive.cli.beeline.BeeLine
   -u <database url>               the JDBC URL to connect to
   -c <named url>                  the named JDBC URL to connect to,
                                   which should be present in beeline-site.xml
                                   as the value of beeline.hs2.jdbc.url.<namedUrl>
   -r                              reconnect to last saved connect url (in conjunction with !save)
   -n <username>                   the username to connect as
   -p <password>                   the password to connect as
   -d <driver class>               the driver class to use
   -i <init file>                  script file for initialization
   -e <query>                      query that should be executed
   -f <exec file>                  script file that should be executed
   -w (or) --password-file <password file>  the password file to read password from
   --hiveconf property=value       Use value for given property
   --hivevar name=value            hive variable name and value
                                   This is Hive specific settings in which variables
                                   can be set at session level and referenced in Hive
                                   commands or queries.
   --property-file=<property-file> the file to read connection properties (url, driver, user, password) from
   --color=[true/false]            control whether color is used for display
   --showHeader=[true/false]       show column names in query results
   --escapeCRLF=[true/false]       show carriage return and line feeds in query results as escaped \r and \n
   --headerInterval=ROWS;          the interval between which heades are displayed
   --fastConnect=[true/false]      skip building table/column list for tab-completion
   --autoCommit=[true/false]       enable/disable automatic transaction commit
   --verbose=[true/false]          show verbose error messages and debug info
   --showWarnings=[true/false]     display connection warnings
   --showDbInPrompt=[true/false]   display the current database name in the prompt
   --showNestedErrs=[true/false]   display nested errors
   --numberFormat=[pattern]        format numbers using DecimalFormat pattern
```

        b. **`hive`** untuk mengetahui versi dari Hive yang terinstal pada HDP sandbox

```
[root@sandbox-hdp ~]# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder
.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://sandbox-hdp.hortonworks.com:2181/default;password=hive;serviceDiscoveryMode=zooKeeper;user=hive;zoo
KeeperNamespace=hiveserver2
23/05/08 15:17:07 [main]: INFO jdbc.HiveConnection: Connected to sandbox-hdp.hortonworks.com:10000
Connected to: Apache Hive (version 3.1.0.3.0.1.0-187)
Driver: Hive JDBC (version 3.1.0.3.0.1.0-187)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.0.3.0.1.0-187 by Apache Hive
```

3. Lakukan input sampeldata (csv/parquet/avro) dari yang tersedia ke dalam HDFS (bisa dilihat kembali mekanismenya di modul 2 dan modul 6).

```
[root@sandbox-hdp ~]# hdfs dfs -mkdir tryhive
[root@sandbox-hdp ~]# hdfs dfs -put "sampeldata.csv" "user/root/tryhive"
```

Hasil file yang sudah tersimpan kedalam HDFS

```
[root@sandbox-hdp ~]# hdfs dfs -ls tryhive
Found 1 items
-rw-r--r--   1 root hdfs       1420 2023-05-08 16:16 tryhive/sampeldata.csv
```

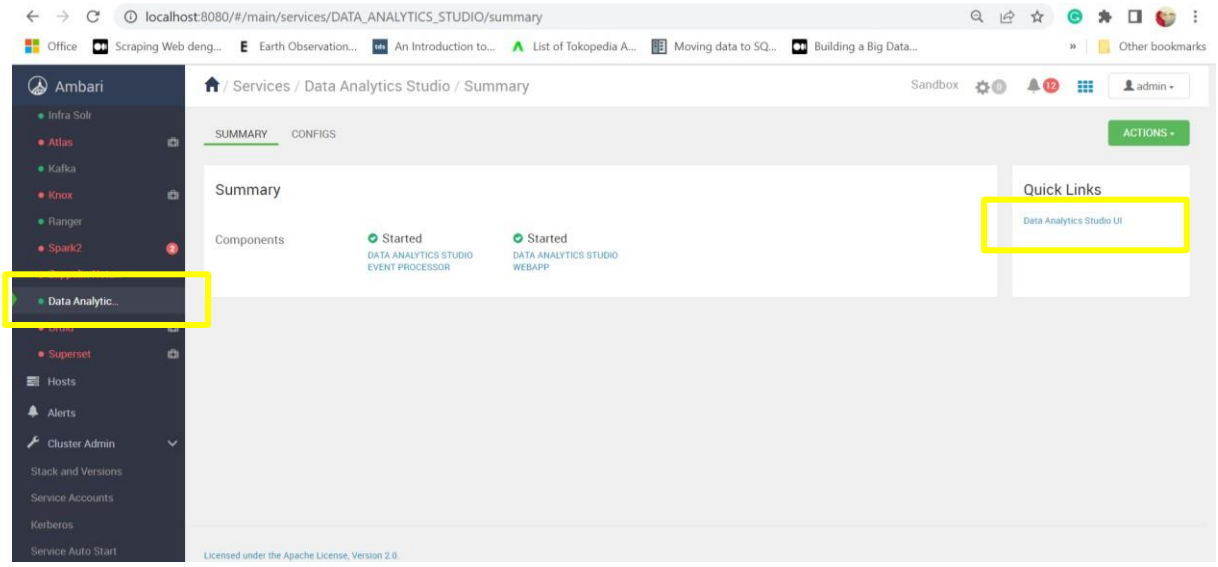
4. Lakukan input sampeldata kedalam folder **/user/hive** dengan memberi akses user root ke folder /user/hive/

```
[root@sandbox-hdp ~]# sudo -u hdfs hadoop fs -chown root /user/hive
[root@sandbox-hdp ~]# hadoop fs -put "sampeldata.csv" "/user/hive/sampeldata.csv"
[root@sandbox-hdp ~]# hadoop fs -ls /user/hive
Found 5 items
drwxr-xr-x   - hive hdfs          0 2018-11-29 19:04 /user/hive/{hive_metastore_warehouse_dir}
drwxr-xr-x   - hive hdfs          0 2018-11-29 17:56 /user/hive/.hiveJars
drwxr-xr-x   - hive hdfs          0 2023-05-09 01:54 /user/hive/jobs
drwxr-xr-x   - hive hdfs          0 2023-02-07 03:42 /user/hive/repl
-rw-r--r--   1 root hdfs       1420 2023-05-09 03:33 /user/hive/sampeldata.csv
```

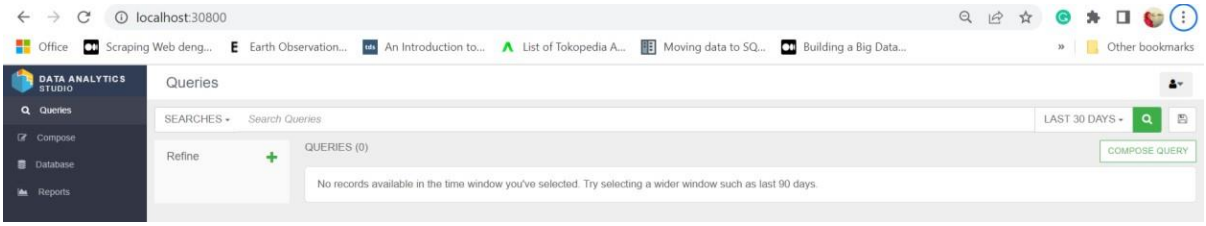
B. Analisis Data menggunakan DAS UI

Ada dua cara melakukan query menggunakan Hive yaitu menggunakan DAS UI dan CLI. Syntax yang digunakan akan sama.

1. Buka Jendela DAS UI, dengan cara klik pada Service **Data Analytics…**



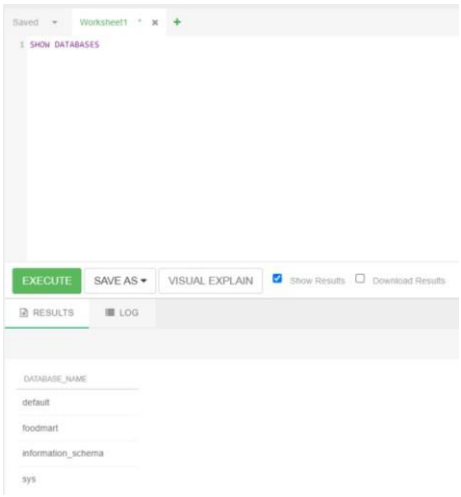
2. Kemudian pada Quick Links, klik **Data Analytics Studio UI**. Maka akan membuka tab browser baru dengan alamat http://sandbox-hdp.hortonworks.com:30800/. Lakukan penggantian alamat ini menjadi http://localhost:30800/, kemudian Enter. Maka akan muncul Jendela DAS.



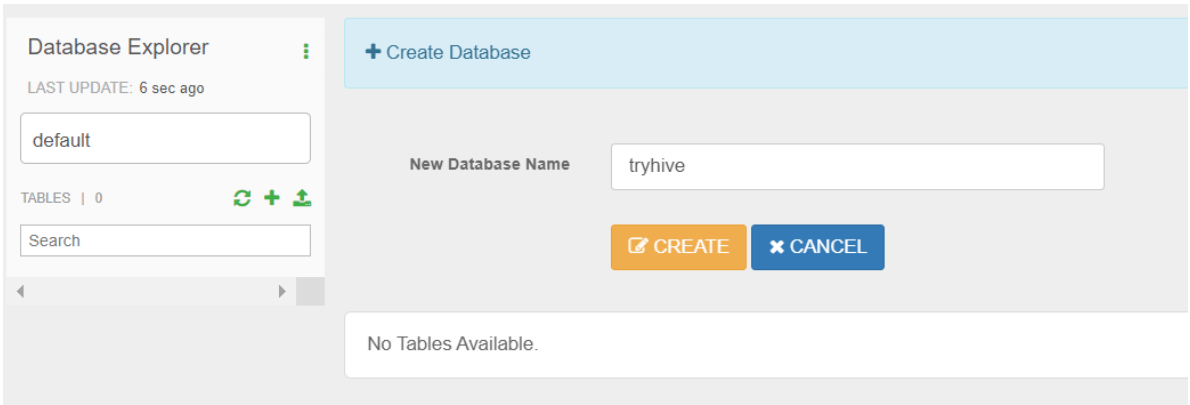
3. Untuk melakukan berbagai query data, klik pada button **Compose Query**. Perintah untuk menampilkan keseluruhan database default yang ada dalam HDP adalah

```
SHOW DATABASES
```

4. Maka pada bingkai **Result** akan menampilkan list database yang tersedia



5. Untuk membuat Database baru, dapat di klik tab **Database** di menu sebelah kiri, kemudian klik Create Database. Misalnya dalam gambar ini adalah membuat database yang bernama “tryhive”.



6. Untuk mengeksekusi query pada database, kita bisa menggunakan:

```
USE [nama database];
[query];
```

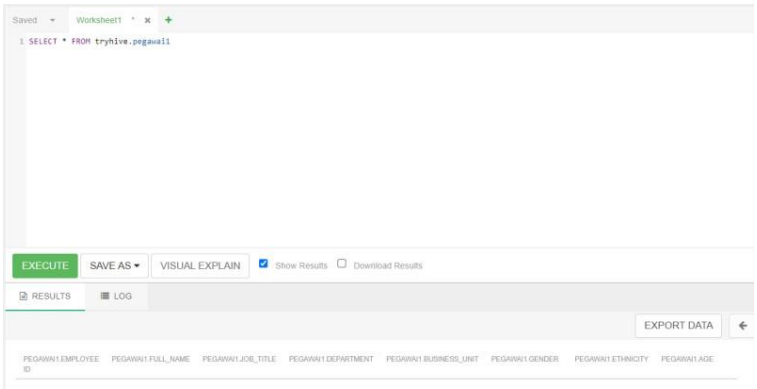
dan diikuti dengan query yang akan dieksekusi atau bisa menuliskan langsung nama database sebelum nama tabel pada query yang akan dieksekusi:

```
[nama database].[nama tabel]
```

7. Untuk membuat Tabel baru (*managed table*), pada bingkai query> Worksheet1. Ketik query sebagai berikut. Untuk struktur skema tabel, dapat menyesuaikan dengan data yang akan di input.

```
CREATE TABLE IF NOT EXISTS `[nama database].[nama tabel]` (
  `Employee ID` varchar(100) NOT NULL,
  `Full_Name` varchar(100) NOT NULL,
  `Job_Title` varchar(100) NOT NULL,
  `Department` varchar(100) NOT NULL,
  `Business_Unit` varchar(100) NOT NULL,
  `Gender` varchar(100) NOT NULL,
  `Ethnicity` varchar(100) NOT NULL,
  `Age` varchar(100) NOT NULL
)
COMMENT 'Tabel Pegawai'
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
STORED AS TEXTFILE;
```

8. Hasilnya adalah skema tabel sebagai berikut



9. Selanjutnya kita dapat melakukan insert data menggunakan syntax INSERT SQL.

```
INSERT INTO `[nama database].[nama tabel]` VALUES
('1', 'Kai Le', 'Controls Engineer', 'Engineering',
'Manufacturing', 'Male', 'Asian', '47'),
('2', 'Robert Patel', 'Analyst', 'Sales', 'Corporate',
'Male', 'Asian', '58'),
('3', 'Cameron Lo', 'Network Administrator', 'IT', 'Research
& Development', 'Male', 'Asian', '34');
```

10. Untuk melihat hasil datanya dapat menggunakan SQL SELECT

```
SELECT * FROM `tryhive.pegawai1`
```

11. Untuk melakukan load data dari file yang sudah di-*store* di HDFS dapat gunakan perintah sebagai berikut:

```
LOAD DATA INPATH '/user/hive/sampeldata.csv' OVERWRITE INTO
TABLE [nama tabel]
```

Karena pada DAS secara default menggunakan user **hive**, maka kita akan load dari dari file data HDFS di folder /user/hive. OVERWRITE artinya kita akan menghapus data yang ada dan menggantinya dengan data baru (optional).

Untuk melakukan load data dari file di local system dapat tambahkan LOCAL pada perintah menjadi:

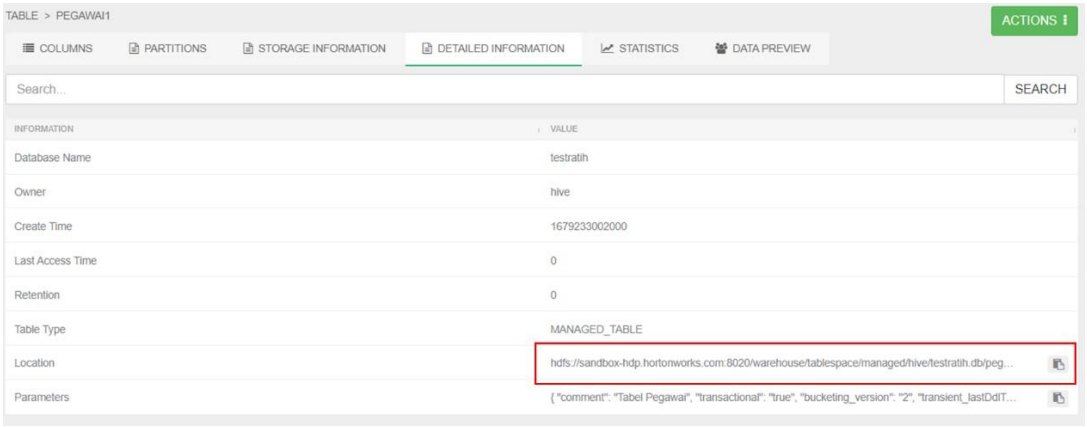
```
LOAD DATA LOCAL INPATH '/home/hive/sampeldata.csv' OVERWRITE
INTO TABLE [nama tabel]
```

Perintah lengkapnya sebagai berikut:

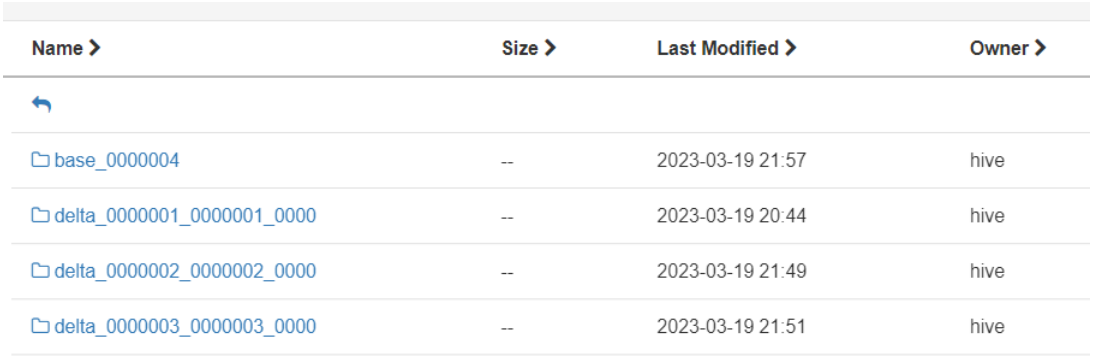
```
LOAD DATA [LOCAL] INPATH 'filepath' [OVERWRITE]
INTO TABLE tablename [PARTITION (partcol1=val1, partcol2=val2
...)] [INPUTFORMAT 'inputformat' SERDE 'serde']
```

Perintah LOAD DATA juga bisa digunakan bersamaan ketika CREATE TABLE.

12. Lokasi dari table tersebut dapat dilihat pada atribut **Location** di Tab **Detailed Information Table**.



File View Ambari:



13. External Table dimana Hive hanya akan mengelola schema-nya, kita buat dulu folder HDFS yang nantinya akan diisi dengan file-file data (karena file data tidak dikelola Hive).

    Misal kita buat folder **ext**:

```
[hive@sandbox-hdp root]$ hdfs dfs -mkdir ext
[hive@sandbox-hdp root]$ hdfs dfs -ls
Found 6 items
drwxr-xr-x   - hive hdfs          0 2018-11-29 19:04 {hive_metastore_warehouse_dir}
drwxr-xr-x   - hive hdfs          0 2018-11-29 17:56 .hiveJars
drwxr-xr-x   - hive hdfs          0 2023-03-19 17:18 ext
drwxr-xr-x   - hive hdfs          0 2023-03-19 17:16 jobs
drwxr-xr-x   - hive hdfs          0 2023-03-19 17:18 repl
```

    Masukan file data sampeldata.csv ke dalam folder ext.

14. Untuk membuat external table kita dapat menggunakan perintah sebagai berikut (perhatikan bahwa atribut kolom tidak boleh mengandung constraint dan location harus berupa directory):

```
CREATE EXTERNAL TABLE IF NOT EXISTS `[nama database].[nama tabel]` (
  `Employee ID` STRING,
  `Full_Name` STRING,
  `Job_Title` STRING,
```

```
   `Department` STRING,
   `Business_Unit` STRING,
   `Gender` STRING,
   `Ethnicity` STRING,
   `Age` STRING
)
COMMENT 'Tabel Pegawai External'
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/hive/ext/';
```

15. Perhatikan lokasi dari external table tersebut yang dapat dilihat pada atribut Location di Tab Detailed Information Table.

C. Analisis Data menggunakan CLI

Cara kedua untuk menjalankan CLI adalah dengan akses ke http://localhost:4200 pada browser atau menggunakan aplikasi MobaXterm.

1. Untuk menjalankan perintah Hive, dapat menggunakan syntax **hive.**

```
[root@sandbox-hdp ~]# hive
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder
.class]
SLF4J: Found binding in [jar:file:/usr/hdp/3.0.1.0-187/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.
class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://sandbox-hdp.hortonworks.com:2181/default;password=hive;serviceDiscoveryMode=zooKeeper;user=hive;zoo
KeeperNamespace=hiveserver2
23/05/08 20:09:10 [main]: INFO jdbc.HiveConnection: Connected to sandbox-hdp.hortonworks.com:10000
Connected to: Apache Hive (version 3.1.0.3.0.1.0-187)
Driver: Hive JDBC (version 3.1.0.3.0.1.0-187)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.0.3.0.1.0-187 by Apache Hive
```

2. Maka akan mundul koneksi ke jdbc:hive. Tuliskan kembali syntax seperti pada bagian B. Untuk melihat database yang tersedia, gunakan syntax **SHOW DATABASES;**

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> SHOW DATABASES;
INFO  : Compiling command(queryId=hive_20230508201450_f0ab8797-293c-49e9-a461-b25cf4c5668b): SHOW DATABASES
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:database_name, type:string, comment:from deserializer)],
properties:null)
INFO  : Completed compiling command(queryId=hive_20230508201450_f0ab8797-293c-49e9-a461-b25cf4c5668b); Time taken: 0.051 secon
ds
INFO  : Executing command(queryId=hive_20230508201450_f0ab8797-293c-49e9-a461-b25cf4c5668b): SHOW DATABASES
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20230508201450_f0ab8797-293c-49e9-a461-b25cf4c5668b); Time taken: 0.059 secon
ds
INFO  : OK
+----------------------+
|    database_name     |
+----------------------+
| default              |
| foodmart             |
| information_schema   |
| sys                  |
| tryhive              |
+----------------------+
5 rows selected (0.471 seconds)
```

3. Untuk melihat tabel yang ada di database dapat gunakan perintah **SHOW TABLES;**

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> SHOW TABLES;
INFO  : Compiling command(queryId=hive_20230508201557_a92637cf-f1c5-4470-b29d-9f6fbb128105): SHOW TABLES
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:tab_name, type:string, comment:from deserializer)], prope
rties:null)
INFO  : Completed compiling command(queryId=hive_20230508201557_a92637cf-f1c5-4470-b29d-9f6fbb128105); Time taken: 0.047 secon
ds
INFO  : Executing command(queryId=hive_20230508201557_a92637cf-f1c5-4470-b29d-9f6fbb128105): SHOW TABLES
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20230508201557_a92637cf-f1c5-4470-b29d-9f6fbb128105); Time taken: 0.032 secon
ds
INFO  : OK
+-----------+
| tab_name  |
+-----------+
| pegawai1  |
+-----------+
1 row selected (0.117 seconds)
```

4. Lakukan untuk insert data.

```
0: jdbc:hive2://sandbox-hdp.hortonworks.com:2> INSERT INTO `pegawai1` VALUES ('1', 'Kai Le', 'Controls Engineer', 'Engineering
', 'Manufacturing', 'Male', 'Asian', '47');
INFO  : Compiling command(queryId=hive_20230508202019_beb1a095-e054-4408-8eab-229f2236b993): INSERT INTO `pegawai1` VALUES ('1
', 'Kai Le', 'Controls Engineer', 'Engineering', 'Manufacturing', 'Male', 'Asian', '47')
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:_col0, type:varchar(100), comment:null), FieldSchema(name
:_col1, type:varchar(100), comment:null), FieldSchema(name:_col2, type:varchar(100), comment:null), FieldSchema(name:_col3, ty
pe:varchar(100), comment:null), FieldSchema(name:_col4, type:varchar(100), comment:null), FieldSchema(name:_col5, type:varchar
(100), comment:null), FieldSchema(name:_col6, type:varchar(100), comment:null), FieldSchema(name:_col7, type:varchar(100), com
ment:null)], properties:null)
INFO  : Completed compiling command(queryId=hive_20230508202019_beb1a095-e054-4408-8eab-229f2236b993); Time taken: 1.205 secon
ds
INFO  : Executing command(queryId=hive_20230508202019_beb1a095-e054-4408-8eab-229f2236b993): INSERT INTO `pegawai1` VALUES ('1
', 'Kai Le', 'Controls Engineer', 'Engineering', 'Manufacturing', 'Male', 'Asian', '47')
INFO  : Query ID = hive_20230508202019_beb1a095-e054-4408-8eab-229f2236b993
INFO  : Total jobs = 1
INFO  : Launching Job 1 out of 1
INFO  : Starting task [Stage-1:MAPRED] in serial mode
INFO  : Subscribed to counters: [] for queryId: hive_20230508202019_beb1a095-e054-4408-8eab-229f2236b993
INFO  : Tez session hasn't been created yet. Opening session
```

5. Untuk perintah lainnya sama dengan perintah eksekusi query di bagian B.

   Gunakan Ctrl+C untuk keluar dari hive shell.

1.5 Penugasan

Load data kedalam HIVE untuk data dengan beberapa format yang berbeda yaitu sampeldata.csv, sampel_data.parquet dan sampel_data.avro. Lakukan beberapa kueri sederhana. **Hasil pekerjaan praktikum berupa dokumentasi atau screenshot Proses dan Hasil kueri data ke HIVE.**