filkom.ub.id

DIGITAL TALENT SCHOLARSHIP 2019

Big Data Analytics

Hadoop Part. 2
(Map Reduce, Oozie dan Sqoop)

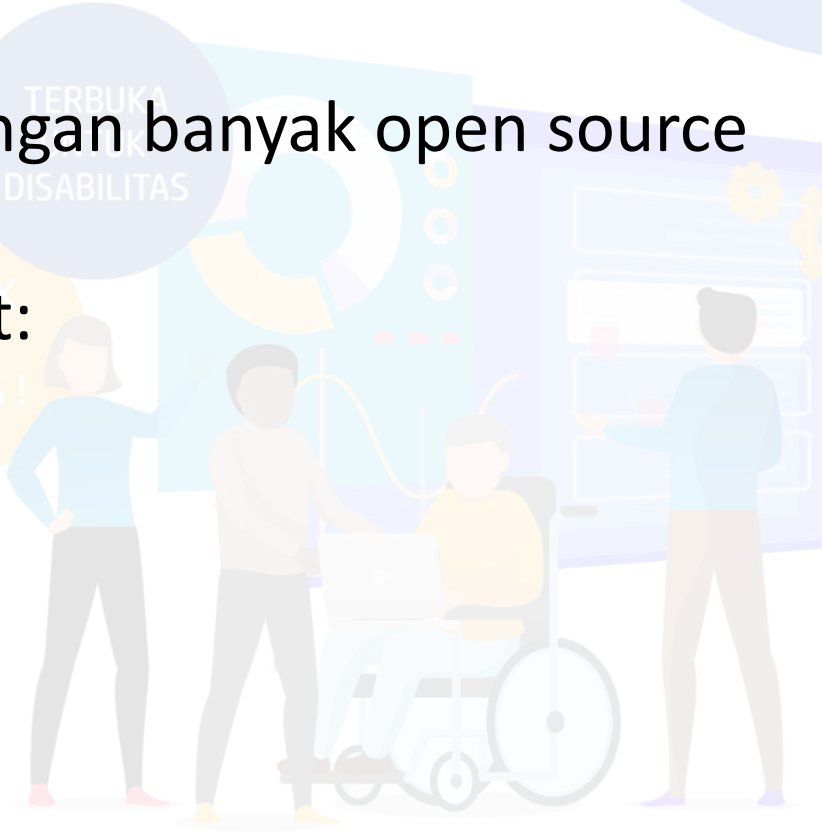
Oleh: Imam Cholissodin | imamcs@ub.ac.id, Putra Pandu Adikara, Sufia Adha Putri
Asisten: Guedho, Sukma, Anshori, Aang dan Gusti
Fakultas Ilmu Komputer (Filkom) Universitas Brawijaya (UB)

Pokok Pembahasan:

- Memahami filosofi MapReduce & Contohnya
- Mempelajari scheduling dan kontrol eksekusi job dengan Oozie
- Sqoop:
  - ✓Apa itu Sqoop
  - ✓Bagaimana cara kerjanya

Pendahuluan

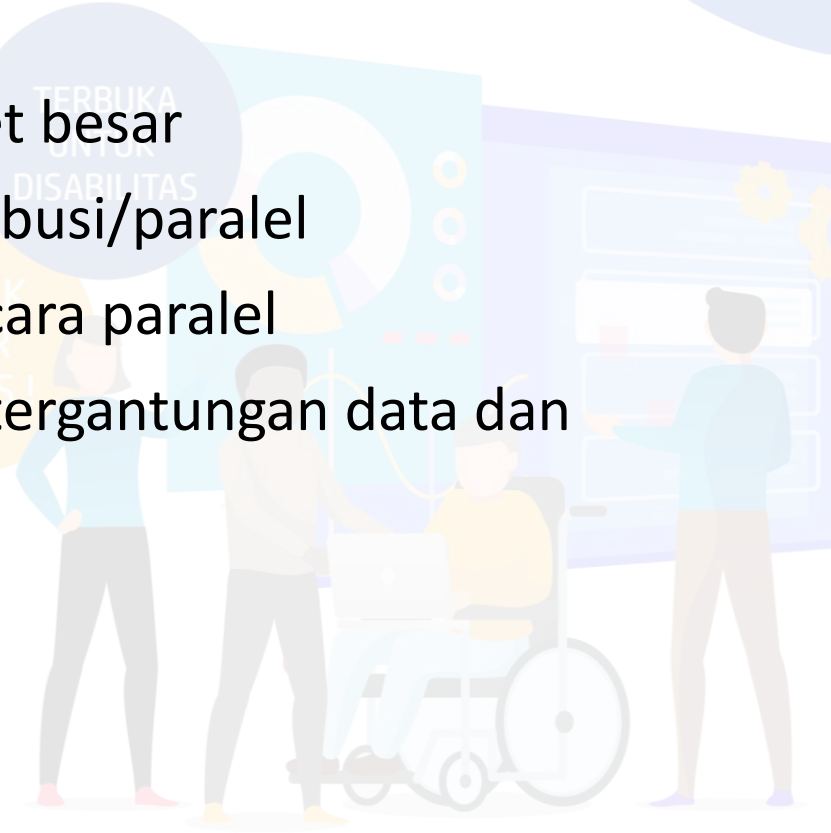
- Environment Hadoop diisi dengan banyak open source project
- Beberapa open source project:
  - **MapReduce**
  - **Oozie**
  - **Hive**
  - **Flume**
  - **Sqoop**
  - **Pig**
  - **dll**



MapReduce

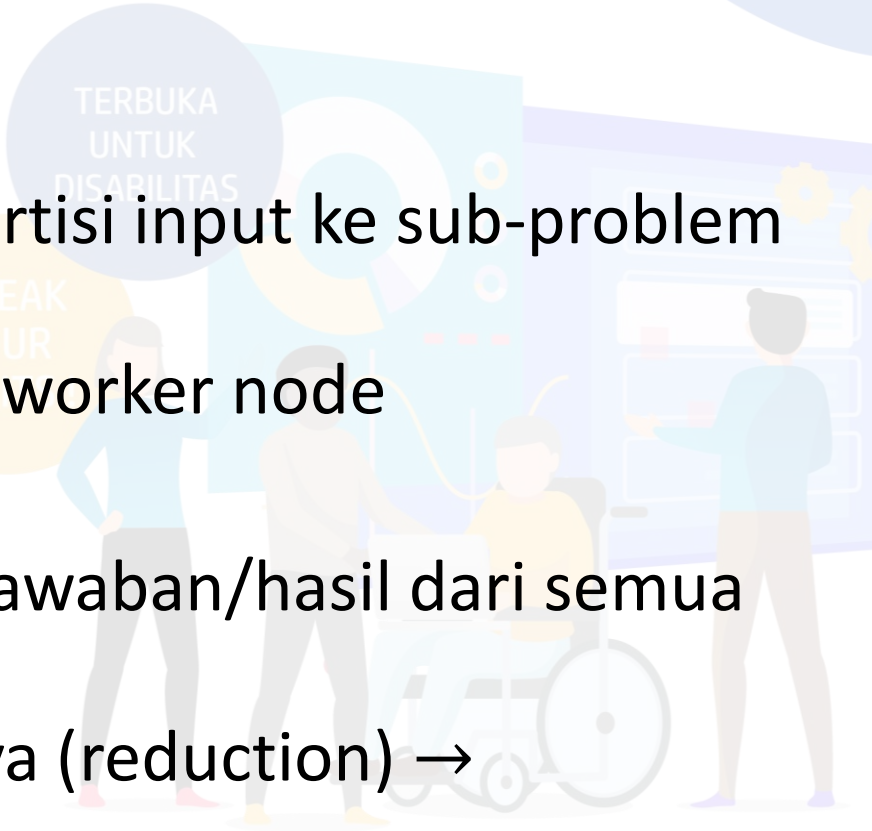
Review MapReduce

- Didesain untuk memproses dataset besar
- Khusus masalah yang bisa terdistribusi/paralel
- Proses disebar ke banyak node secara paralel
- Tidak boleh ada dependencies/ketergantungan data dan proses

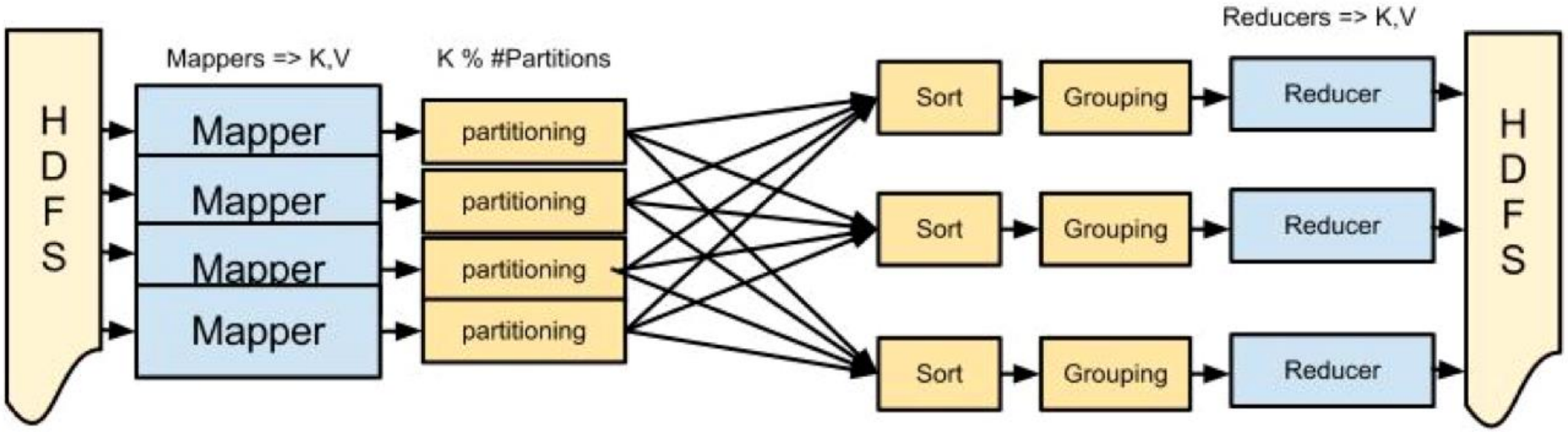


Review MapReduce

- Map
  - Master node membagi/partisi input ke sub-problem lebih kecil
  - Distribusi sub-problem ke worker node
- Reduce
  - Master node mengambil jawaban/hasil dari semua sub-problem
  - Menggabungkan semuanya (reduction) → hasil/output
- Map dan Reduce ← distributed processing



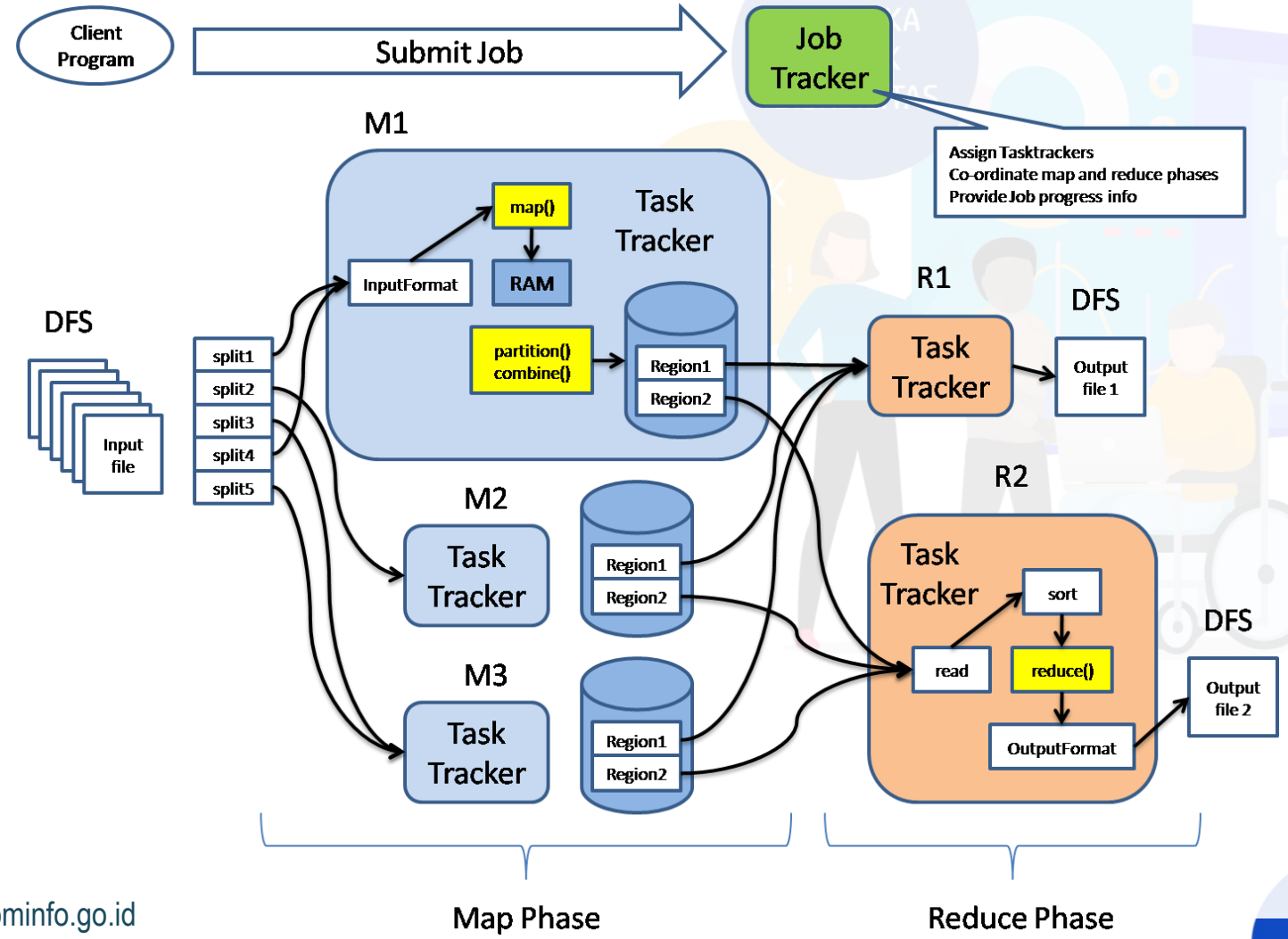
Proses MapReduce



The MapReduce Pipeline

A mapper receives (Key, Value) & outputs (Key, Value)
A reducer receives (Key, Iterable[Value]) and outputs (Key, Value)
Partitioning / Sorting / Grouping provides the Iterable[Value] & Scaling

Sebuah job dengan 1 step map dan 1 step reduce

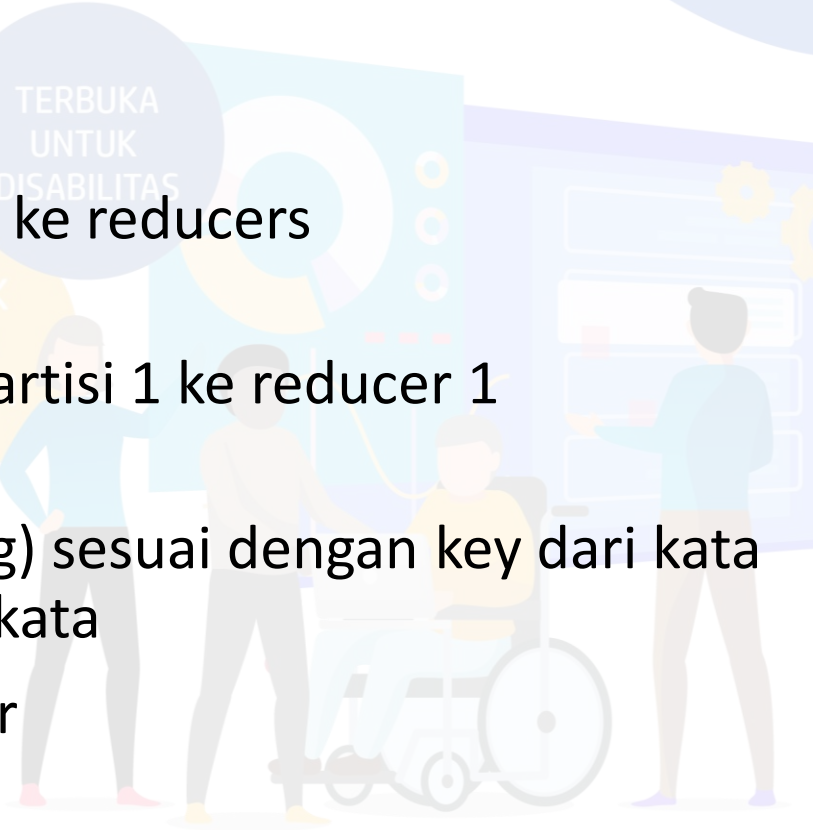


List Proses - Map

- Map step
  - Input split → Mengambil subset (sebagian) data input dari full dataset
  - Operasi dilakukan ke tiap baris dari input split (tergantung operasinya, parsing dll)
  - Outputnya di-*buffered* dalam memori dan taruh ke disk
  - Di-*sort* dan dipartisi oleh key dengan default partitioner
  - Merge sort → urut tiap partisi
  - Bisa ada beberapa map secara paralel → proses input split beda

List Proses - Reduce

- Reduce step
  - ➢Partisi dari output map di-*shuffle* ke reducers
    Partisi 1 ke reducer 1
  - ➢Jika ada beberapa map, semua partisi 1 ke reducer 1
    Partisi 2 ke reducer 2, dst
  - ➢Melakukan proses merge (gabung) sesuai dengan key dari kata
    contoh: jumlah kemunculan tiap kata
  - ➢Hasilnya diurutkan di tiap reducer

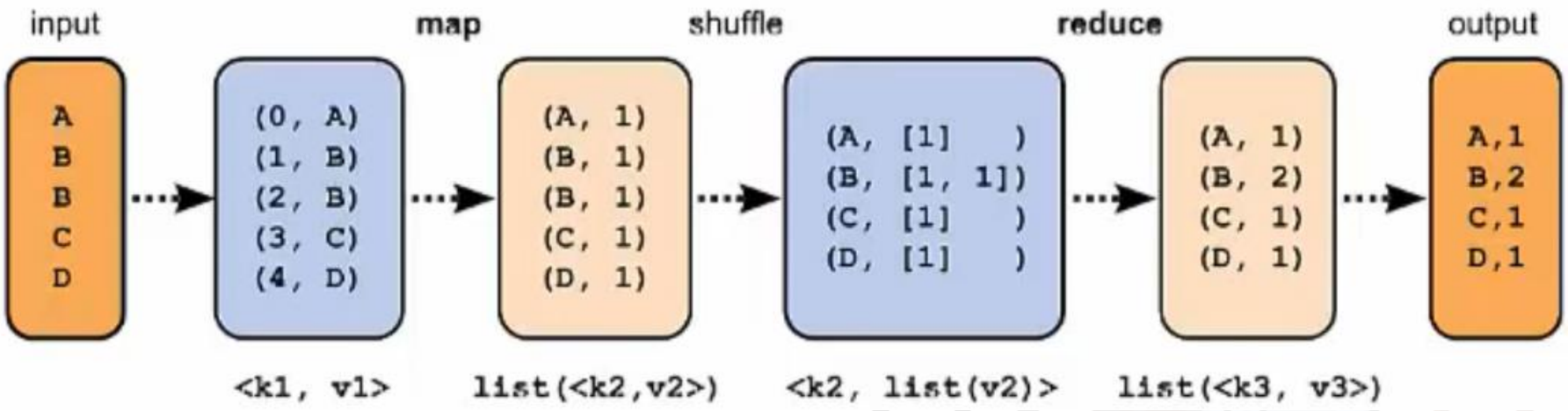


Fundamental Data Type

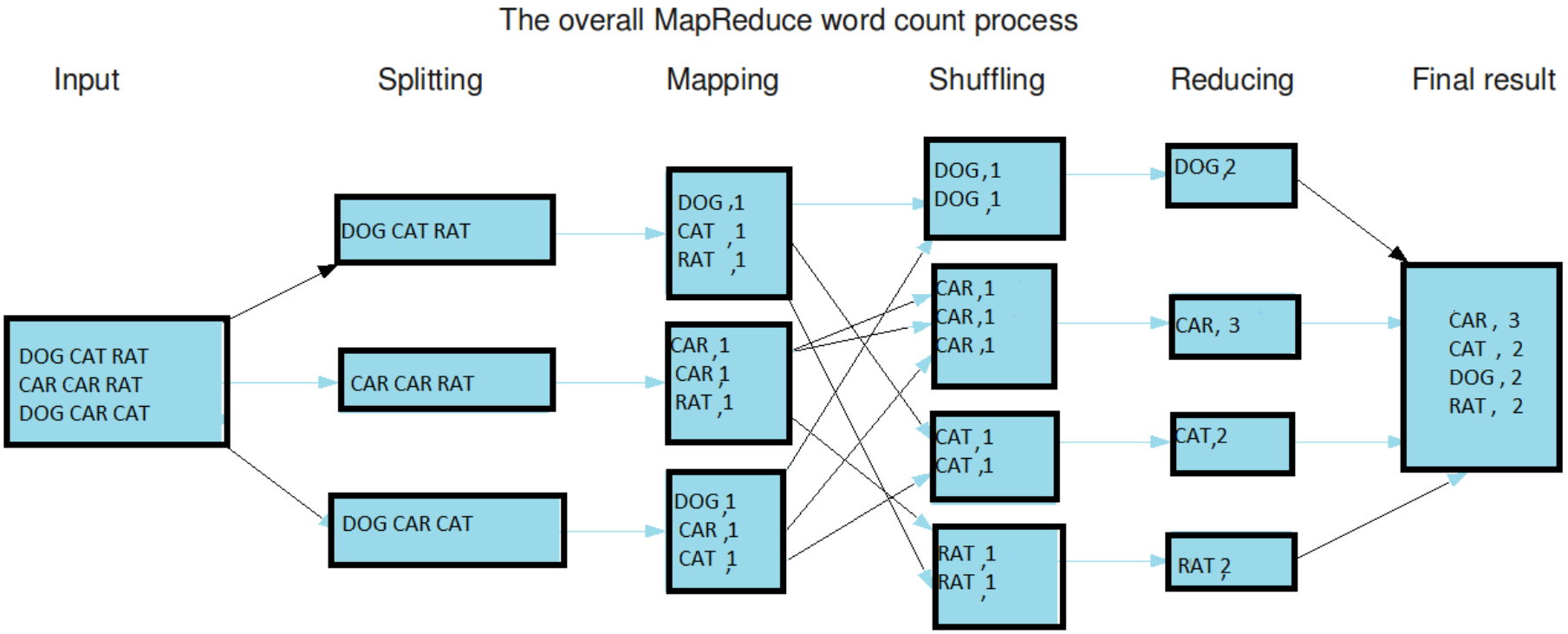
- Data MapReduce yang masuk dan keluar dalam bentuk unstructured
- Sebelum masuk ke Hadoop, diubah ke key-value pair Hadoop menyuplai key-nya
  - Key-value
  - List

	Input	Output
Map	<k1, v1>	List(<k2, v2>)
Reduce	<k2, list(v2)>	List(<k3, v3>)

Contoh Key-Value dan List



Word Counter



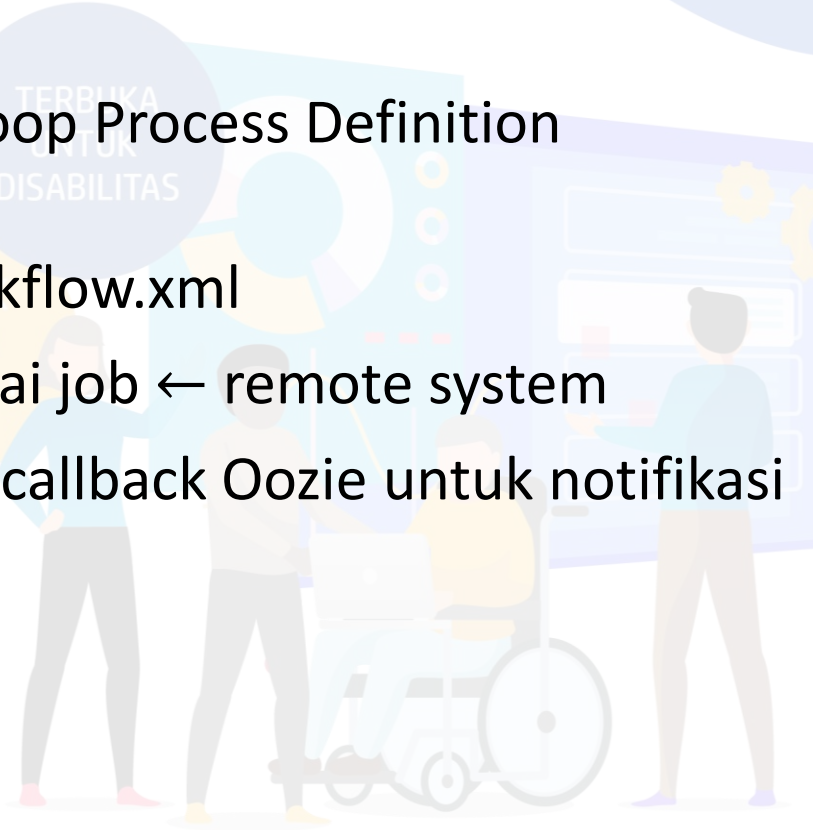
Oozie

Oozie (1)

- Komponen open source Hadoop job control
- Menangani Hadoop jobs
- Oozie workflow → kumpulan action yang diatur dalam DAG (Direct Acyclic Graph)
- Ada control dependency (kebutuhan kendali) dalam 1 aksi ke aksi berikutnya
  Contoh:
  Proses berikutnya tidak bisa berjalan hingga proses sebelumnya selesai
  Proses final/penggabungan hanya bisa dijalankan ketika semua proses paralel sebelumnya selesai

Oozie (2)

- Workflow ditulis dalam hPDL (Hadoop Process Definition Language) → sebuah XML PDL
- Disimpan dalam sebuah file → workflow.xml
- Tiap workflow action untuk memulai job ← remote system
- Action complete → remote system callback Oozie untuk notifikasi

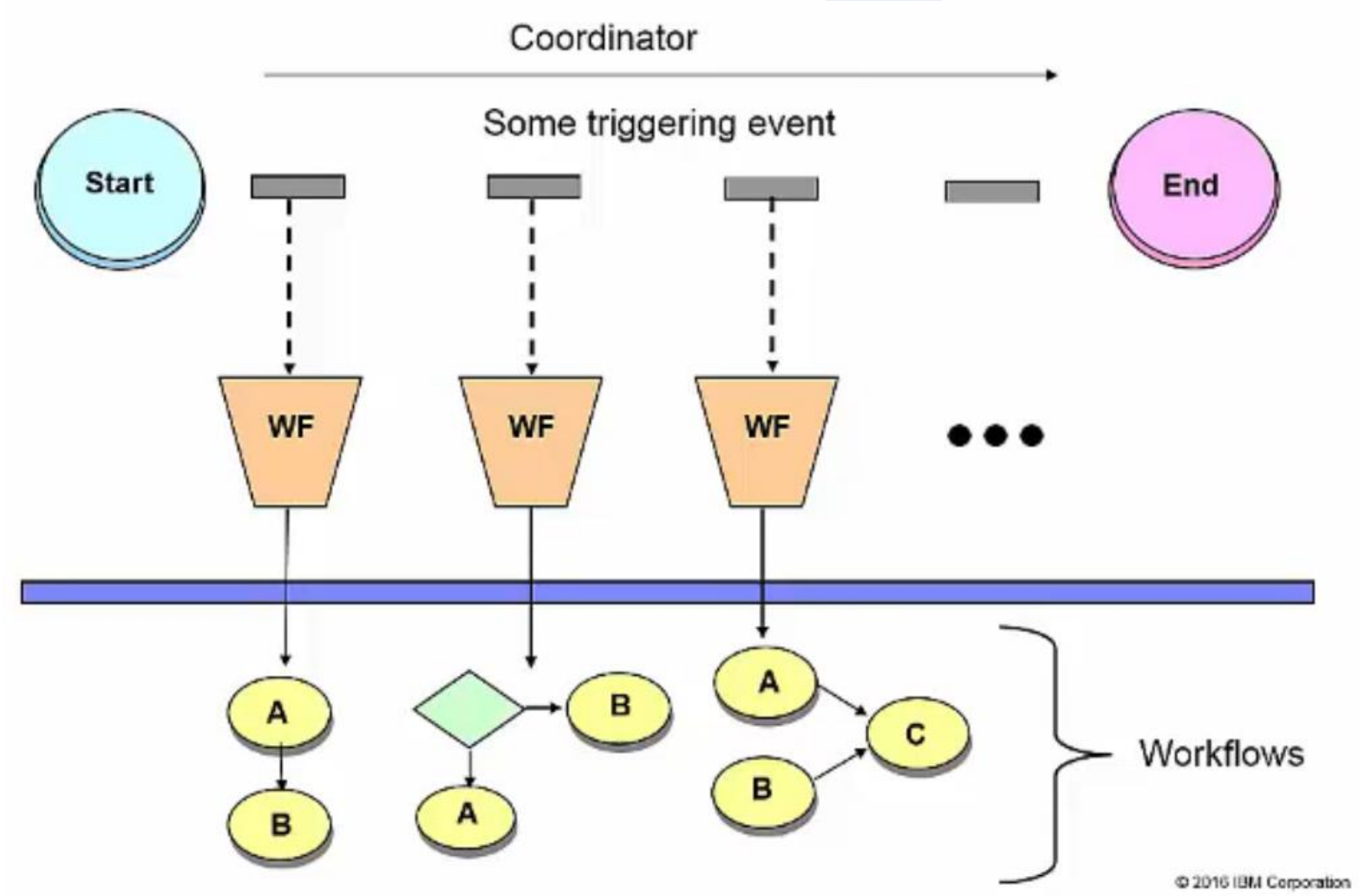


Oozie Coordinator

- Komponen koordinator bisa panggil workflow berdasar interval waktu (contoh tiap 15 menit) atau berdasarkan ketersediaan data.
- Connect workflow jobs yang berjalan reguler namun interval berbeda. →PENTING

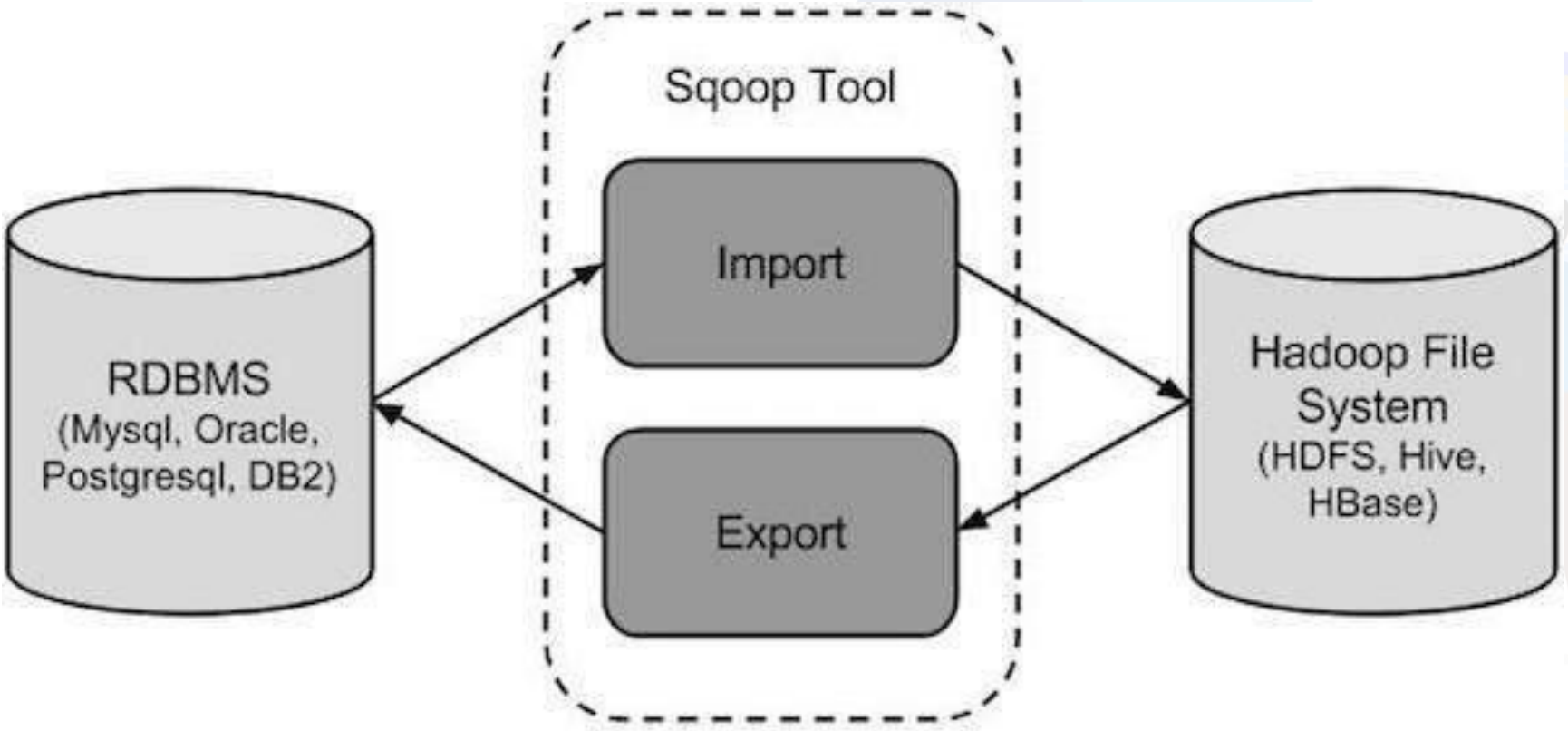
  Contoh: Output dari 4 job terakhir (@15 menit) jadi input dari job (@1jam).
- Satu workflow bisa panggil 1/beberapa task baik sekuensial atau berdasar control logic.

Oozie Coordinator (2)



Sqoop

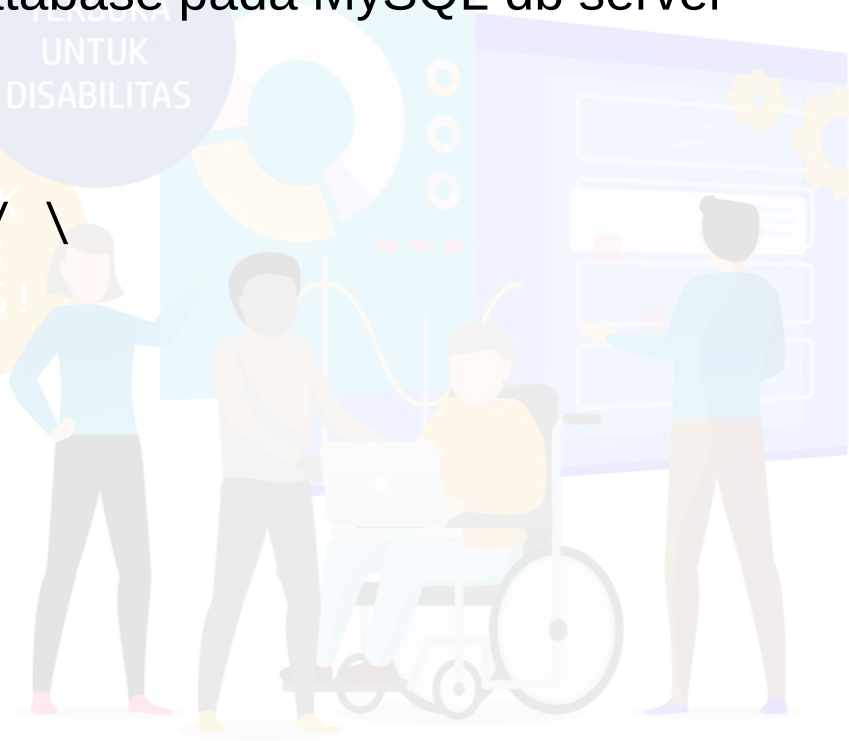
Cara Kerja Sqoop



Perintah pada Sqoop

- List Database, menampilkan daftar database pada server
  - Contoh untuk menampilkan list database pada MySQL db server

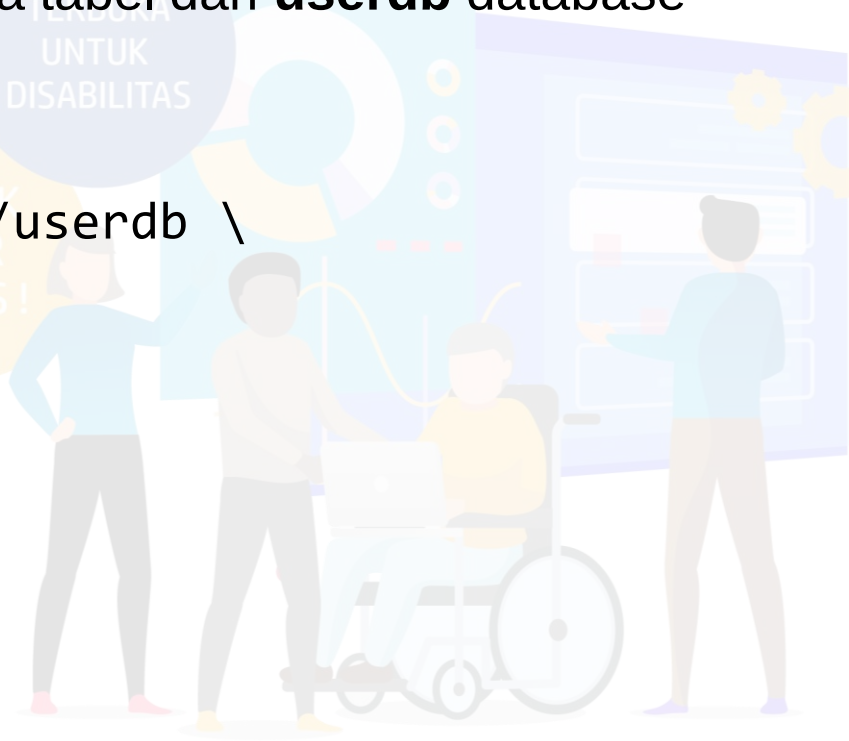
```
$ sqoop list-databases \
--connect jdbc:mysql://localhost/ \
--username root
```



Perintah pada Sqoop

- List Table, menampilkan daftar table dari database
  - Contoh untuk menampilkan semua tabel dari **userdb** database

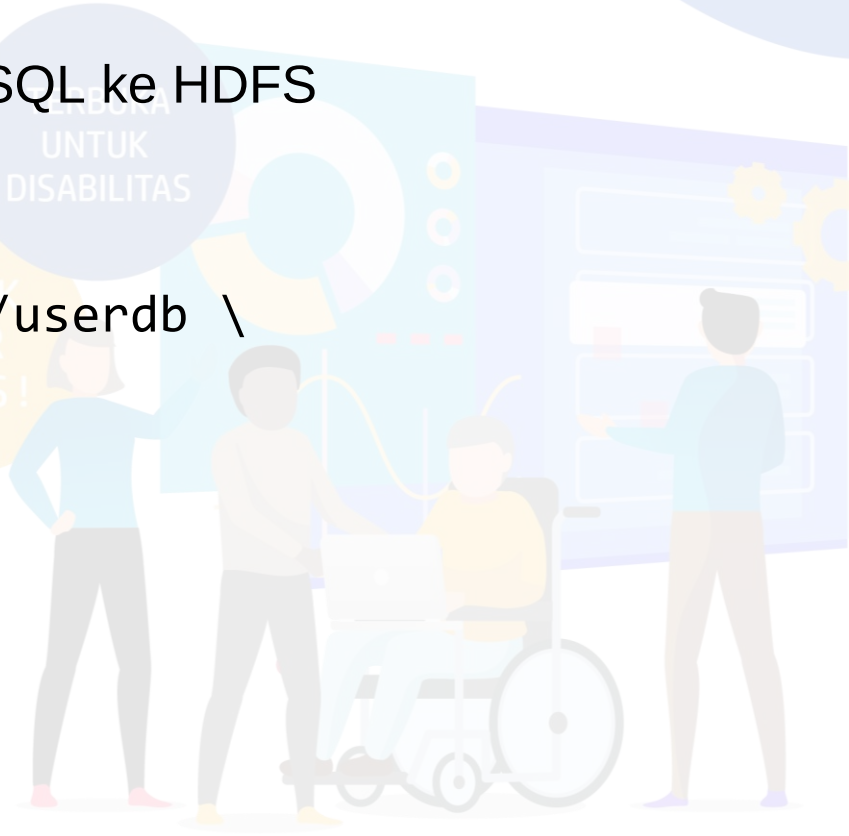
```
$ sqoop list-tables \
--connect jdbc:mysql://localhost/userdb \
--username root
```



Perintah pada Sqoop

- Import Table
  - Contoh import tabel **emp** dari MySQL ke HDFS

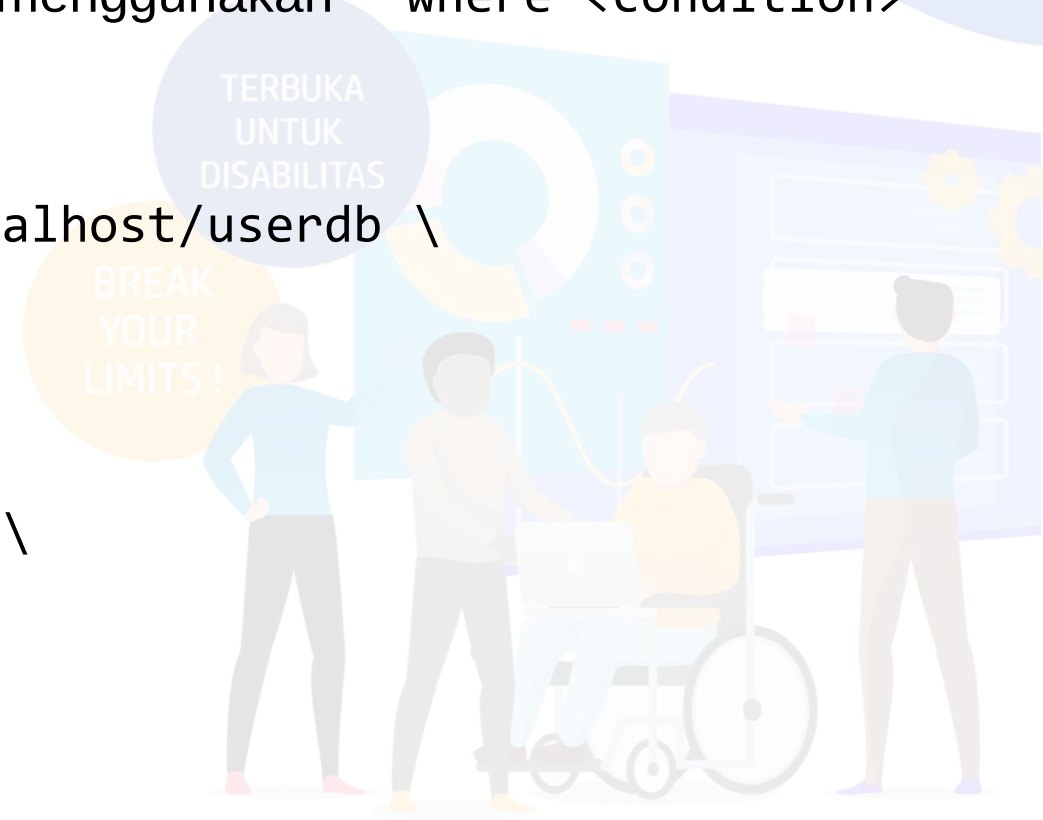
```
$ sqoop import \
--connect jdbc:mysql://localhost/userdb \
--username root \
--table emp --m 1
```



Perintah pada Sqoop

- Import table dengan kondisi, menggunakan `--where <condition>`

```
$ sqoop import \
--connect jdbc:mysql://localhost/userdb \
--username root \
--table emp_add \
--m 1 \
--where “city =’sec-bad’” \
--target-dir /wherequery
```



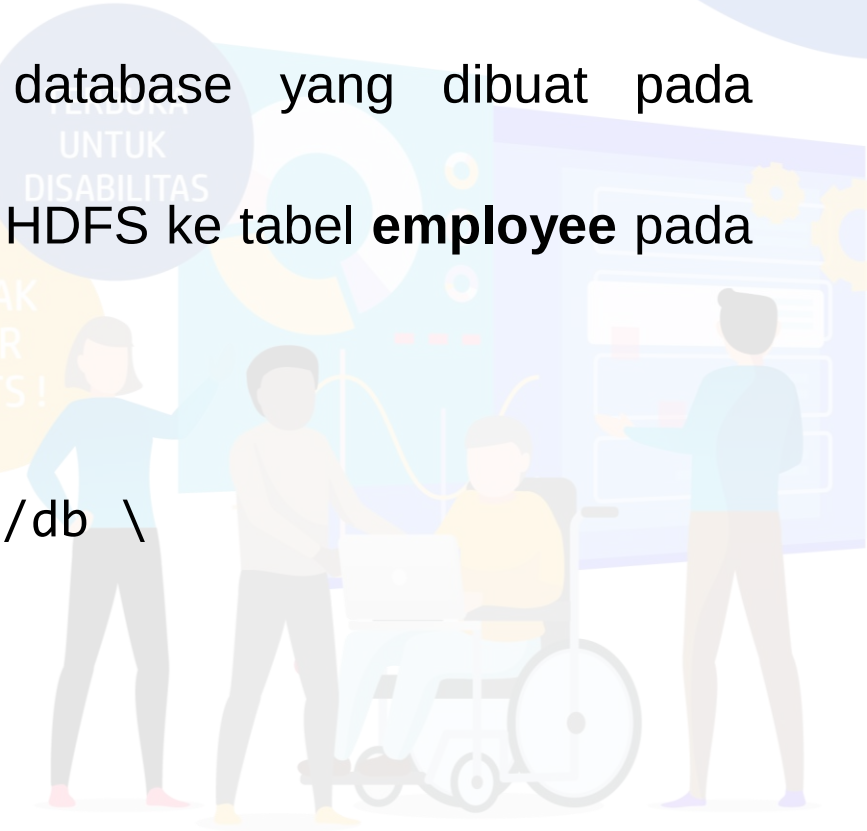
Perintah pada Sqoop

- Import semua tabel pada suatu database
  - Contohnya import semua tabel daru **userdb** database :

```
$ sqoop import-all-tables \
--connect jdbc:mysql://localhost/userdb \
--username root
```

Perintah pada Sqoop

- Export dari HDFS ke RDBMS
  - Syaratnya harus sudah ada database yang dibuat pada database server.
  - Contoh : export **emp_data** pada HDFS ke tabel **employee** pada MySQL database server




```
$ sqoop export \
--connect jdbc:mysql://localhost/db \
--username root \
--table employee \
--export-dir /emp/emp_data
```

Perintah pada Sqoop

- Mengecek isi pada HDFS

```
$ hadoop fs -ls
```

Sqoop Job

- Contoh : membuat job bernama **myjob** untuk import data dari tabel **employee** ke HDFS

```
$ sqoop job --create myjob \
-- import \
--connect jdbc:mysql://localhost/db \
--username root \
--table employee --m 1
```

Sqoop Job

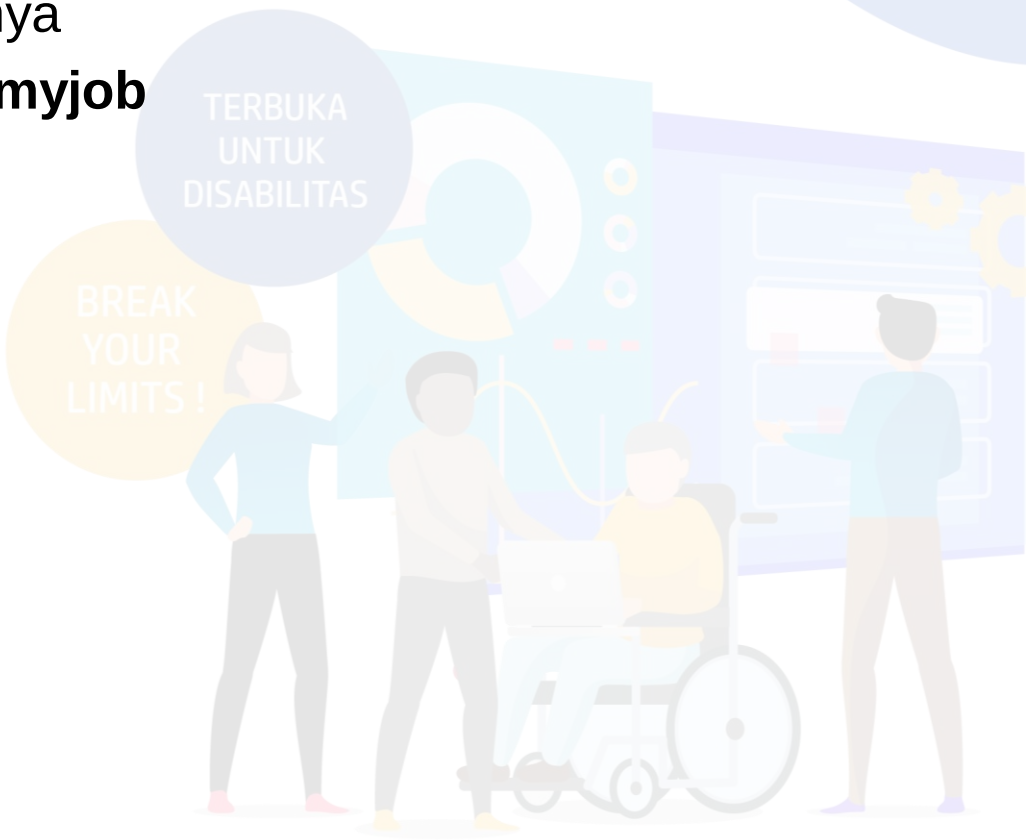
- Melihat daftar Sqoop Jobs

```
$ sqoop job --list
```

Sqoop Job

- Untuk inspect jobs dan detailnya
  - Contohnya menginspeksi **myjob**

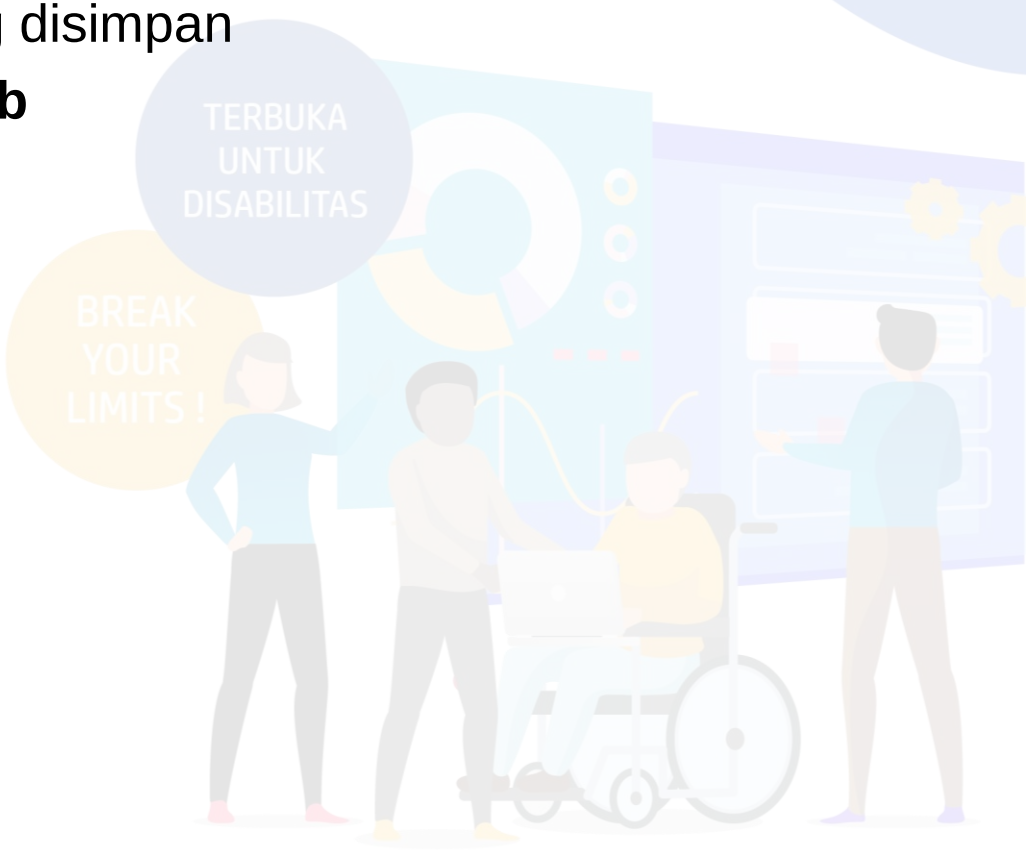
```
$ sqoop job --show myjob
```



Sqoop Job

- Untuk mengeksekusi job yang disimpan
  - Contohnya eksekusi **myjob**

```
$ sqoop job --exec myjob
```



Install Sqoop

- Requirements : Hadoop, Java, Database Connector
- Install Sqoop -> https://www.apache.org/dist/sqoop/

Latihan langsung di Kelas Ke-1 & Pembahasan

- Tidak Ada Latihan Ke-1

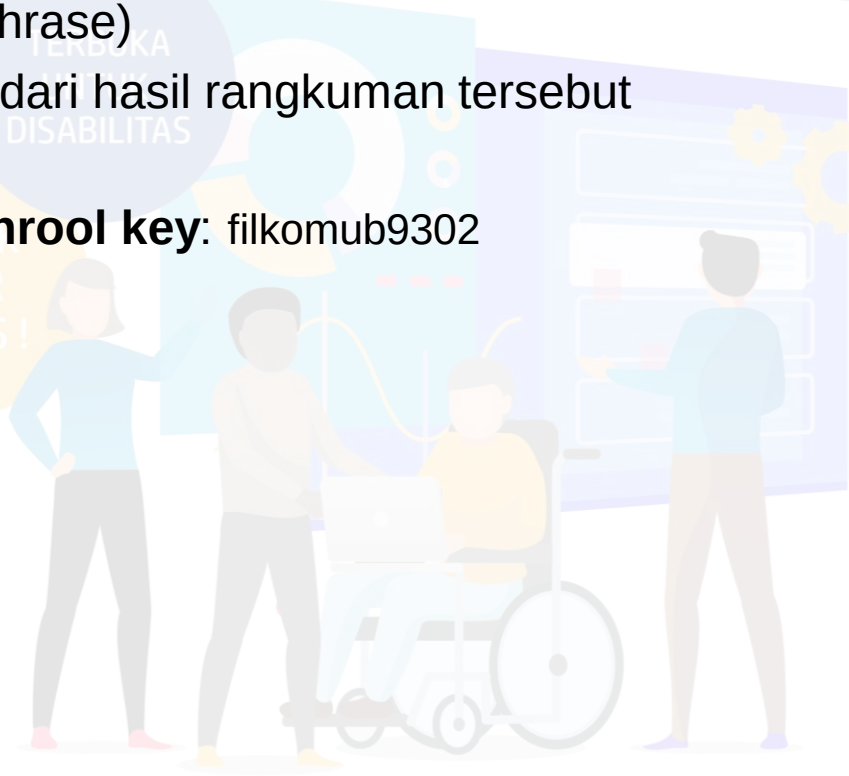
Latihan langsung di Kelas Ke-2 & Pembahasan

- Tidak Ada Latihan Ke-2

Tugas Individu

1. Buatlah rangkuman materi dengan cara berikut:
   - Rangkum yang pokok (bukan paraphrase)
   - Cek plagiasi diturnitin (tiap minggu) dari hasil rangkuman tersebut

     > Register ke turnitin
     > Masukkan **id class**: 21563495 & **enrool key**: filkomub9302



filkom.ub.id

DIGITAL TALENT SCHOLARSHIP 2019

Big Data Analytics

Terimakasih

Oleh: Imam Cholissodin | imamcs@ub.ac.id, Putra Pandu Adikara, Sufia Adha Putri
Asisten: Guedho, Sukma, Anshori, Aang dan Gusti
Fakultas Ilmu Komputer (Filkom) Universitas Brawijaya (UB)