BY CHARLIE'S ANGELS

LEARNING PROGRESS REVIEW

WEEK 10

DigitalSkola

NoSQL

NoSQL adalah sistem manajemen data non-relasional yang tidak memerlukan skema tetap. Istilah ini sendiri berasal dari singkatan “non-SQL” atau “not only SQL.”

Sistem database NoSQL mencakup lebih banyak teknologi database yang bisa menyimpan data secara terstruktur, semi terstruktur, tidak terstruktur, dan juga polimorfik.

Fungsi utama dalam menggunakan database NoSQL adalah sebagai penyimpanan data terdistribusi dimana kebutuhan penyimpanan data sangat besar.

Jenis-Jenis Database NoSQL

1. **Column-based Database**
   Cara ini data disimpan di dalam kolom-kolom.
2. **Graph-oriented Database**
   Data disimpan dalam bentuk graf.
3. **Document Database**
   Cara penyimpanan basis data NoSQL dalam dokumen-dokumen.
4. **Key-value Database**
   cara penyimpanan dalam bentuk kunci – isi berpasangan.

Document Database

Data berisi sepasang key dan value yang disimpan dalam dokumen dengan format JSON atau XML dipakai pada jenis database satu ini. Value yang dimaksud bisa berupa banyak hal, seperti string, angka, boolean, arrays, hingga object. Penggunaan jenis database document-oriented sangat fleksibel sehingga umum digunakan untuk CMS, platform blogging, analisis real-time, dan aplikasi e-commerce. Amazon SimpleDB, CouchDB, dan MongoDB merupakan beberapa contoh database document-oriented yang populer.

Gambar 1 Document JSON yang menyimpan informasi tentang user bernama Tom.

MongoDB

MongoDB adalah salah satu database berjenis NoSQL yang dikembangkan dengan bahasa pemrograman C++. Sistem database ini dapat digunakan oleh para developer untuk mengembangkan berbagai jenis aplikasi dan website yang bersifat terukur (scalable).

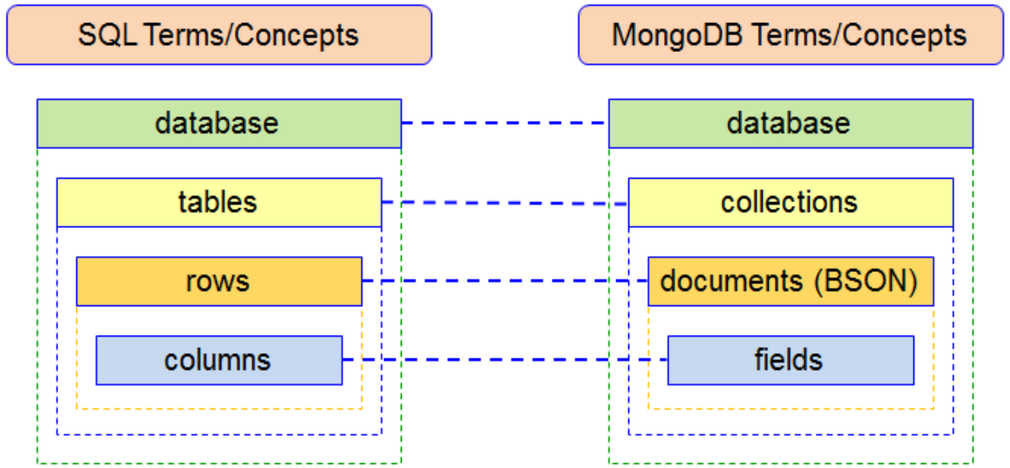
Penyimpanan data dalam format JSON tidak memerlukan struktur tabel atau penyusunan data yang terperinci, sehingga tidak perlu memikirkan tentang relasi antar tabel. Tak heran jika beberapa perusahaan besar seperti Uber, Lyft, Accenture, dan Forbes menggunakan teknologi MongoDB untuk database mereka.

Tiga komponen penting dalam sistem database MongoDB:

Database
Keseluruhan penyimpanan yang berisi susunan data (collection).

Collection
Susunan data yang tersimpan dalam bentuk dokumen (seperti tabel pada sistem database SQL).

Document
Unit data satuan yang ada pada sebuah sistem database MongoDB.



Gambar 2 Persamaan istilah yang dapat dibandingkan antara sistem database SQL dan NoSQL.

Query pada MongoDB

Exporting data JSON format

mongoexport –db database_name –collection collection_name –out path_or_name_of_the_file

Import JSON files

mongoimport –jsonArray –db database_name –collection collection collection_name –file file_location

find() Method

db.collection.find(query, projection)
db.collection.findAndModify(document)
db.collection.findOneAndDelete(filter, options)
db.collection.findOneAndReplace(filter, replacement, options)
db.collection.findOneAndUpdate(filter, update, options)

Update() Method

db.collection.update(query, update, options)
db.collection.updateOne(filter, update, options)
db.collection.updateMany(filter, update, options)

delete() Method

db.collection.deleteOne()
db.collection.deleteMany()

DigitalSkola

ADVANCED CRUD OPERATIONS

MQL OPERATORS

- Query and Projection Operators
  - Query operators untuk mencari dan menemukan data dalam database, projection operators untuk memodifikasi bagaimana data akan ditampilkan
- Update Operators
  - Untuk memodifikasi data atau penambahan data dalam database
- Semua operator diawali dengan simbol $

QUERY OPERATORS - COMPARISON

DigitalSkola

{<field>: {<operator>:<value>}}

Syntax	Fungsi
$eq	Sama dengan (==)
$gt	Lebih besar dari (>)
$gte	Lebih besar dari atau sama dengan (>=)
$ne	Tidak sama dengan (!=)
$lt	Lebih kecil dari (<)
$lte	Lebih kecil dari atau sama dengan (<=)
$in	Mencocokan dengan semua value tertentu pada array
$nin	Mencocokan dengan semua value yang tidak tersedia pada sebuah array

QUERY OPERATORS - LOGIC

Syntax	Fungsi
$and	Mencocokan dengan semua kondisi
$or	Mencocokan dengan minimal satu kondisi terpenuhi
$nor	Mencocokan dengan semua kondisi yang tidak terpenuhi
$not	Mengembalikan documents yang tidak cocok dengan kondisi

DigitalSkola

kafka INTRODUCTION

DigitalSkola

INTRODUCTION

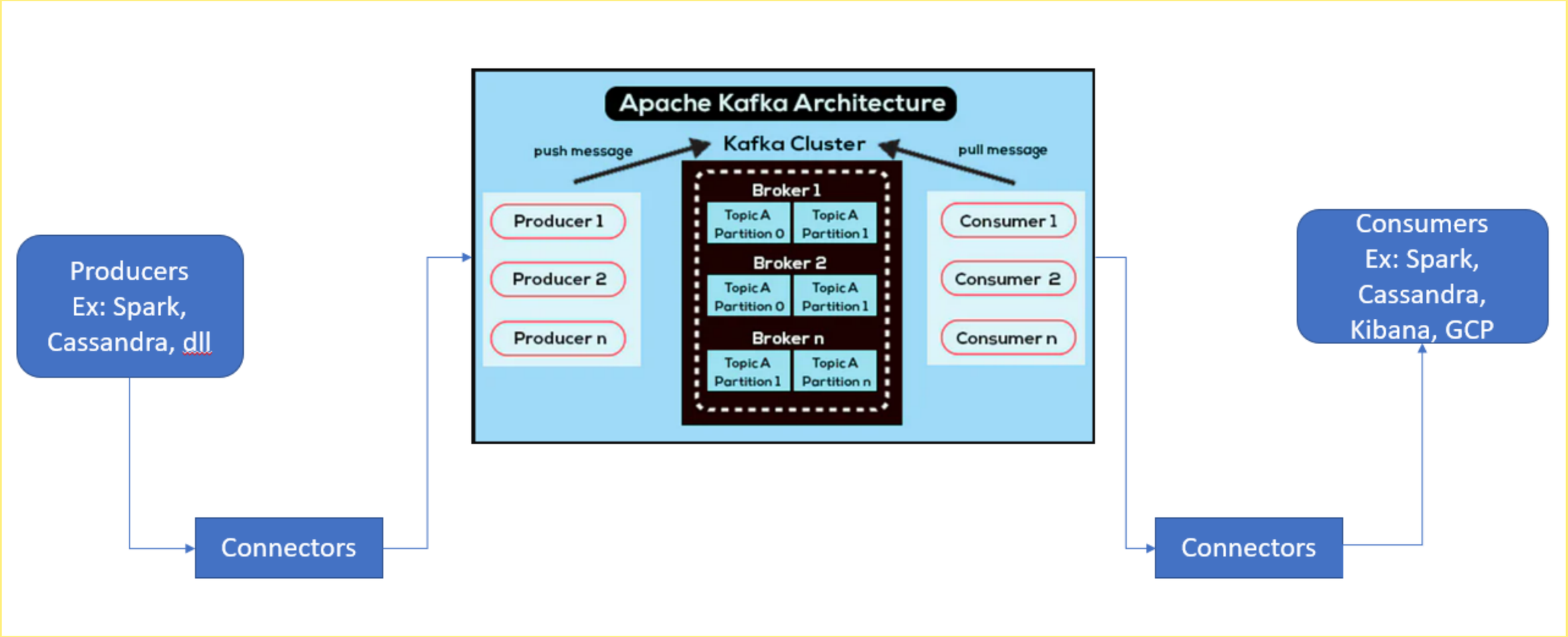
- Kafka dapat didefinisikan sebagai message broker yang terdistribusi yang dirancang untuk proses data streaming (realtime data).
- Kafka akan menerima pesan yang dikirimkan oleh producer dan akan diterima oleh consumer.

KEUNGGULAN KAFKA

- Kafka mendukung multiple producers untuk write data
- Kafka mendukung multiple consumers untuk read data
- Disk Based retention: data akan disimpan dalam disk (7 hari)
- Scalable
- High Perfromance: dapat menangani data dalam jumlah yang sangat besar

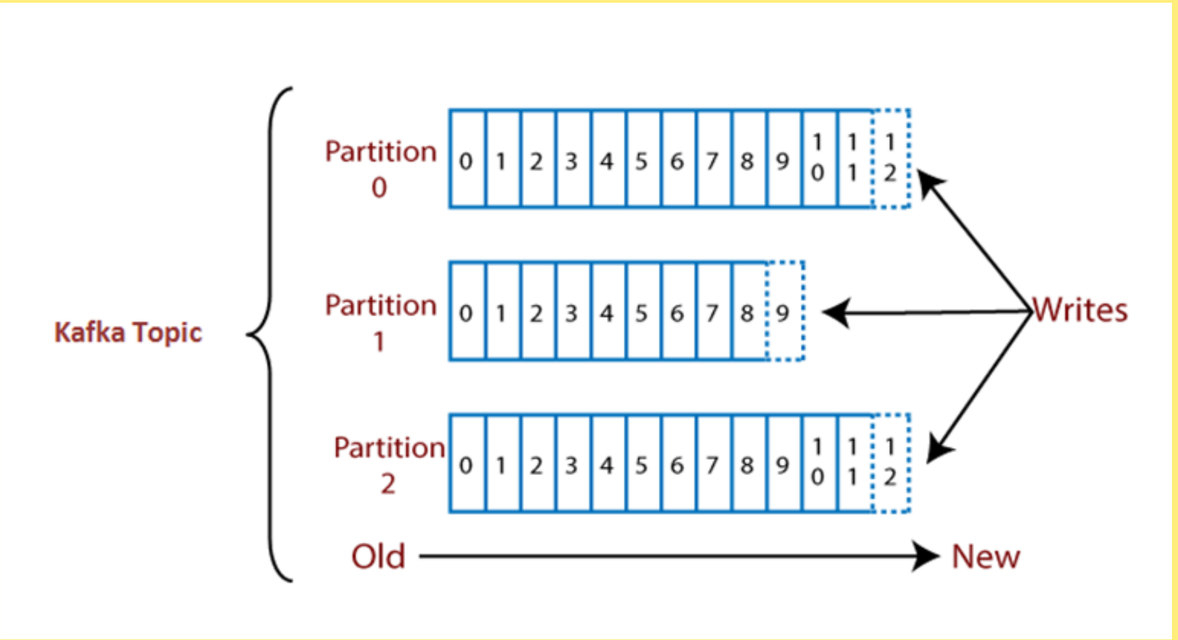
DigitalSkola

KAFKA ARCHITECTURE



KAFKA TOPICS

- Dapat dianalogikan seperti Table pada RDBMS
- Record data akan disimpan pada topics
- Topics akan disimpan kedalam partition
- Setiap record memiliki key, value, dan timestamp
- Record terbaru akan disimpan pada offset selanjutnya



source: static.javatpoint.com

DigitalSkola

KAFKA PRODUCERS

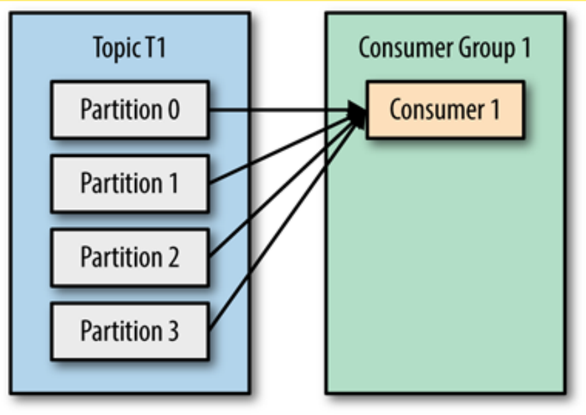
- Publish data kedalam topics tertentu
- Producers bertanggungjawab memilih partisi dan topik yang akan diisi oleh record

KAFKA CONSUMERS

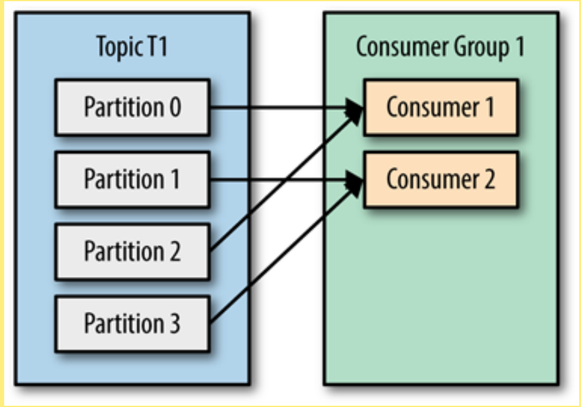
- Consumers dapat dikelompokan menjadi consumers group
- Consumer group dapat men-subscribe pada satu atau lebih topic
- Apabila consumer tidak dikelompokan manjadi consumer group, maka ia akan subscribe setiap partisi.
- Jika dikelompokan menjadi consumer group maka 1 consumer hanya akan subscribe ke partisi yang telah ditentukan

DigitalSkola

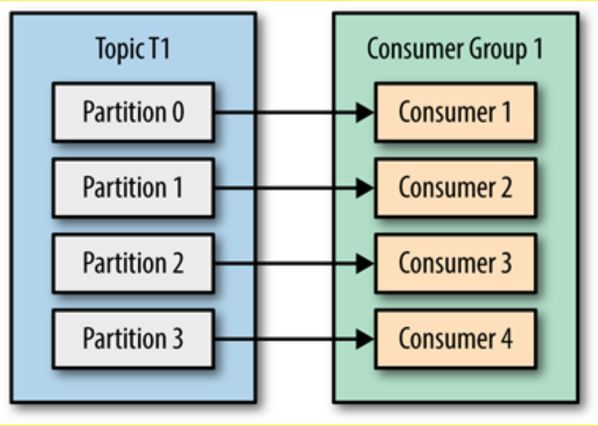
KAFKA CONSUMERS



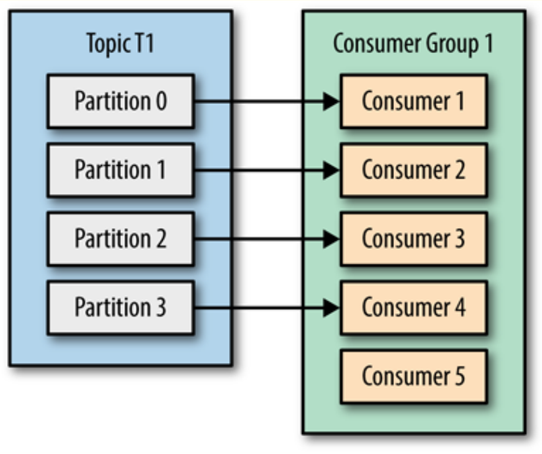
Tanpa consumer group, 1 consumer akan subscribe semua partisi



Dengan 2 consumer pada sebuah consumer group. Partisi dibagi menjadi 2



Dengan full consumer pada sebuah consumer group. Masing-masing consumer subscribe 1 partisi



Lebih banyak consumer pada sebuah consumer group daripada partisi. Consumer ke-5 akan idle tanpa subscribe partisi

source images: oreilly.com

DigitalSkola

KAFKA MESSAGES

- Merupakan isi dari data
- Dianalogikan seperti row pada RDBMS
- Dapat berupa string, csv, json, dan lain-lain

KAFKA DOCKER (STEP BY STEP)

- **Clone github repository untuk install kafka pada docker**
  - https://github.com/wurstmeister/kafka-docker.git
- **Change directory to kafka-docker**
  - cd kafka-docker
- **Update Kafka dan setting docker compose**
  - Nano atau vim file docker-compose.yml
  - Ganti KAFKA_ADVERTISED_HOST_NAME menjadi ip private dari instance atau server yang digunakan
  - Start docker compose : docker-compose up –d
- **Scaling broker (optional)**
  - docker-compose scale kafka=3 (menggunakan 3 broker)

DigitalSkola

MEMBUAT KAFKA TOPIC

- **Masuk ke Kafka shell**
  - ./start-kafka-shell.sh 172.17.0.1
- **Buat topic**
  - $KAFKA_HOME/bin/kafka-topics.sh --create --topic nama_topic \ --partitions 4 --eplication-factor 1 --bootstrap-server `broker-list.sh`
- **Cek apakah topic sudah berhasil dibuat**
  - $KAFKA_HOME/bin/kafka-topics.sh --describe --topic nama_topic \ --bootstrap-server `broker-list.sh`

kafka

MEMBUAT KAFKA PRODUCER

- **Masuk ke Kafka shell**
  - ./start-kafka-shell.sh 172.17.0.1
- **Create console producer**
  - $KAFKA_HOME/bin/kafka-console-producer.sh --topic=nama_topic \ --broker-list=`broker-list.sh`

Terima kasih!

Sampai jumpa di
Learning Progress Review
kami berikutnya!

Hilda Meiranita Prastika Dewi

Nur Indrasari

Rezha Sulvian

Thasha Dinya Ainsha

DigitalSkola